



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - TE 091599

**KONSTRUKSI DIAGRAM *LADDER* MENGGUNAKAN
METODE *FLOW-TABLE/STATE DIAGRAM* UNTUK
SELEKSI DAN PERAKITAN *PART* PADA *DUAL CONVEYOR***

Nicco
NRP 2215105056

Dosen Pembimbing
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 091599

**Construction of Ladder Diagram using Flow-
Table/State Diagram Method for Part Selection and
Assembly on Dual Conveyor**

Nicco
NRP 2215105056

Supervisor
Dr. Ir. Mochammad Rameli
Eka Iskandar, ST., MT.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Electrical Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2017

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Konstruksi Diagram Ladder Menggunakan Metode *Flow-Table/State Diagram* untuk Seleksi dan Perakitan Part pada Dual Conveyor**” adalah merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

Nicco
Nrp 2215 105 056

[Halaman ini sengaja dikosongkan]

**KONSTRUKSI DIAGRAM LADDER MENGGUNAKAN
METODE *FLOW-TABLE/STATE DIAGRAM* UNTUK SELEKSI
DAN PERAKITAN PART PADA *DUAL CONVEYOR***

TUGAS AKHIR

Diajukan untuk Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing I



Dr. Ir. Mochammad Rameli
NIP. 19541212 1981031002

Dosen Pembimbing II



Eka Iskandar, ST., MT.
NIP. 19800528 2008121001



[Halaman ini sengaja dikosongkan]

KONSTRUKSI DIAGRAM *LADDER* MENGUNAKAN METODE *FLOW-TABLE/STATE* *DIAGRAM* UNTUK SELEKSI DAN PERAKITAN *PART* PADA *DUAL CONVEYOR*

Nicco – 2215105056

Pembimbing : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRAK

Di dalam industri, *conveyor* merupakan alat transportasi untuk memindahkan material produksi. *Conveyor* tidak sekedar memindahkan barang saja melainkan terdapat proses lain seperti menghitung jumlah barang, pengisian cairan dan lain sebagainya karena dilengkapi sensor dan aktuator tertentu. Proses-proses tersebut dapat diotomatiskan dengan menggunakan PLC sebagai kontroler dengan menjalankan program instruksi berupa *ladder diagram*. Kompleksnya proses otomasi pada *conveyor* membutuhkan penggunaan *relay* yang banyak dalam pemrograman PLC dan sulit diterapkan pada PLC yang rendah spesifikasinya. Tentunya pengadaan PLC dengan spesifikasi tinggi membutuhkan biaya yang tidak sedikit. Agar dapat mengatasi permasalahan tersebut maka diperlukan suatu metode yang dapat meminimalkan jumlah *relay* yang digunakan sehingga program menjadi lebih singkat. Dengan menggunakan *state diagram*, hasil konstruksi *ladder diagram* diperoleh sebanyak 32 *rung* dan kapasitas program yang dihasilkan sebesar 3 KB.

Kata kunci : *Conveyor, Flow-Table/State Diagram, Ladder Diagram, PLC, Rung.*

[Halaman ini sengaja dikosongkan]

Construction of Ladder Diagram using Flow-Table/State Diagram Method for Part Selection and Assembly on Dual Conveyor

Nicco – 2215105056

Supervisor : 1. Dr. Ir. Mochammad Rameli
2. Eka Iskandar, ST., MT.

ABSTRACT

In the industry, conveyor is a transportation tool that move production materials. Conveyor not just move goods but there are other processes such as counting the quantity of goods, filling the liquid and so forth because it is equipped with certain sensors and actuators. These processes can be automated by using PLC as a controller by executing instruction program in the form of ladder diagram. The complexity of automation processes in conveyors requires the use of many relays in PLC programming and is difficult to apply to low-specification PLCs. Of course the procurement of PLC with high specifications require a lot of money. In order to overcome these problems then it is needed a method that can minimize the amount of relays used so that the program becomes shorter. By using state diagram, the result of ladder diagram construction is obtained 32 rung and the program capacity is 3 KB.

Keywords : Conveyor, Flow-Table/State Diagram, Ladder Diagram, PLC, Rung.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa karena berkat rahmat dan karunia-Nya penulis dapat menyelesaikan tugas akhir yang berjudul **“Konstruksi Diagram Ladder Menggunakan Metode Flow-Table/State Diagram untuk Seleksi dan Perakitan Part pada Dual Conveyor”** Pada kesempatan ini penulis menyampaikan terima kasih atas segala bantuan dan dukungannya yang telah diberikan selama proses pembuatan tugas akhir ini kepada :

1. Orang tua tercinta serta keluarga dan kerabat yang senantiasa memberikan doa serta dukungan.
2. Bapak Mochamad Rameli selaku dosen pembimbing yang selalu memberikan pengarahan, saran dan motivasi dalam kelancaran tugas akhir ini.
3. Bapak Eka Iskandar selaku dosen pembimbing yang telah banyak memberikan ilmu dan bimbingan selama penulis mengerjakan tugas akhir ini.
4. Seluruh dosen, staf dan karyawan di Teknik Elektro Institut Teknologi Sepuluh Nopember.
5. Seluruh teman-teman Lintas Jalur angkatan 2015, terutama mahasiswa Teknik Sistem Pengaturan.
6. Rekan-rekan asisten laboratorium Teknik Sistem Pengaturan.
7. Semua pihak yang telah membantu penulis dalam menyelesaikan tugas akhir ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari masih banyak kekurangan dalam tugas akhir ini. Kritik dan saran untuk perbaikan tugas akhir ini sangat diperlukan. Semoga tugas akhir ini dapat bermanfaat bagi semua pihak.

Surabaya, Juli 2017

Nicco
NRP 2215 105 056

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan	2
1.4 Metodologi	2
1.5 Sistematika	3
1.6 Relevansi	3
BAB 2 DASAR TEORI	5
2.1 <i>Dual Conveyor Workcell Systems</i>	5
2.1.1 <i>Power Supply dan Interface</i>	6
2.1.2 Benda Kerja (<i>Part</i>)	7
2.1.3 <i>Conveyor</i>	8
2.1.4 <i>Dispenser dan Flipper</i>	9
2.1.5 <i>Opto-detector</i>	11
2.1.6 Sensor Induktif	12
2.1.7 <i>Height Detector</i>	14
2.2 <i>Programmable Logic Controller (PLC)</i>	14
2.2.1 Komponen Dasar PLC	17
2.2.2 Bahasa Pemrograman PLC.....	18
2.2.3 Pengalamatan dan Instruksi Dasar PLC	21
2.3 <i>Flow-Table/State Diagram</i>	23
2.3.1 Deskripsi I/O dan <i>State</i>	24
2.3.2 Penyusunan <i>State Diagram</i> (I/O).....	24
2.3.3 <i>Primitive Flow Table</i>	25
2.3.4 <i>Merged Flow Table</i>	29
2.3.5 Penyusunan <i>State Diagram</i> (R/O)	31
2.4 Konstruksi <i>State Diagram</i> (R/O)- <i>Ladder</i>	32
BAB 3 PERANCANGAN SISTEM	35

3.1	Perumusan Sistem <i>Dual Conveyor</i>	35
3.1.1	I/O Sistem	36
3.1.2	Langkah Kerja Sistem.....	38
3.2	Perancangan <i>State Diagram</i>	42
3.2.1	Penyusunan <i>State Diagram</i> (I/O).....	44
3.2.2	Penyusunan <i>Primitive Flow Table</i>	45
3.2.3	Penyusunan <i>Merged Flow Table</i>	45
3.2.4	Penyusunan <i>State Diagram</i> (R/O)	45
3.3	Konstruksi <i>State</i> (R/O)- <i>Ladder</i>	46
BAB 4 PENGUJIAN DAN ANALISA.....		49
4.1	Pengkabelan (<i>Wiring</i>).....	49
4.2	Pengujian Sistem.....	51
BAB 5 KESIMPULAN DAN SARAN		53
5.1	Kesimpulan	53
5.2	Saran	53
DAFTAR PUSTAKA		55
LAMPIRAN		57
RIWAYAT HIDUP		71

DAFTAR GAMBAR

Gambar 2.1	<i>Dual Conveyor Workcell Systems</i>	5
Gambar 2.2	Bagian <i>Dual Conveyor</i>	6
Gambar 2.3	Lokasi <i>Power Supply</i> dan <i>Interface PCB</i>	7
Gambar 2.4	<i>Part</i> untuk <i>assembly</i>	7
Gambar 2.5	<i>Dual Conveyor</i>	8
Gambar 2.6	<i>Dispenser</i> dan <i>Flipper</i>	9
Gambar 2.7	Konstruksi <i>Solenoid</i> Linear tipe Tarik.....	10
Gambar 2.8	<i>Opto-detector</i>	11
Gambar 2.9	Sensor Fotoelektrik Model Reflektif.....	11
Gambar 2.10	Sensor Induktif.....	12
Gambar 2.11	Struktur Sensor Induktif.....	13
Gambar 2.12	Pengukur Ketinggian <i>Part</i> ‘Washer’.....	14
Gambar 2.13	PLC <i>Compact</i> dan <i>Modular</i>	15
Gambar 2.14	PLC Omron CP1E-30.....	16
Gambar 2.15	Komponen Dasar PLC.....	17
Gambar 2.16	<i>Ladder Diagram</i>	19
Gambar 2.17	Contoh Instruksi FBD.....	20
Gambar 2.18	Contoh Instruksi ST.....	20
Gambar 2.19	Contoh Instruksi IL.....	20
Gambar 2.20	Contoh Instruksi SFC.....	21
Gambar 2.21	Aturan Pengalamatan PLC Omron.....	21
Gambar 2.22	<i>State diagram</i>	23
Gambar 2.23	Langkah desain <i>state diagram</i>	23
Gambar 2.24	<i>State Diagram</i> (I/O).....	25
Gambar 2.25	Ilustrasi <i>State</i> Tidak Stabil dan <i>State</i> Stabil.....	26
Gambar 2.26	<i>State Diagram</i> (R/O).....	32
Gambar 2.27	<i>State Diagram</i> (R/O) ke <i>Ladder</i>	33
Gambar 3.1	Tahapan penelitian.....	35
Gambar 3.2	<i>Input/Output</i> Sistem.....	36
Gambar 3.3	<i>State Diagram</i> (I/O) Sub Proses 1.....	44
Gambar 3.4	<i>State Diagram</i> (R/O) Sub Proses 1.....	46
Gambar 3.5	Konstruksi <i>Ladder</i> untuk Sub Proses 1.....	47
Gambar 4.1	Pengkabelan pada PLC.....	50

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Kriteria ketinggian benda kerja.....	8
Tabel 2.2 Keterangan Indikator.....	16
Tabel 2.3 Instruksi Dasar PLC.....	22
Tabel 2.4 Keterangan I/O	24
Tabel 2.5 Terjemahan urutan <i>state diagram</i> (I/O).....	24
Tabel 2.6 Kombinasi bit <i>input</i>	25
Tabel 2.7 <i>State</i> stabil	27
Tabel 2.8 <i>State</i> tidak stabil.....	27
Tabel 2.9 <i>Don't care</i>	28
Tabel 2.10 Kombinasi bit <i>output</i>	28
Tabel 2.11 <i>Primitive Flow Table</i>	29
Tabel 2.12 Kombinasi bit <i>output</i> yang sama.....	29
Tabel 2.13 <i>Merged Flow Table</i> (1)	30
Tabel 2.14 <i>Merged Flow Table</i> (2)	31
Tabel 2.15 <i>Merged Flow Table</i> (3)	31
Tabel 3.1 <i>Input</i> Sistem.....	37
Tabel 3.2 <i>Output</i> Sistem	38
Tabel 3.3 Bit I/O Sistem	42
Tabel 3.4 Urutan <i>State Diagram</i> (I/O).....	43
Tabel 3.5 <i>Primitive Flow Table</i> Sub Proses 1	45
Tabel 3.6 <i>Merged Flow Table</i> Sub Proses 1	45
Tabel 4.1 Alamat I/O PLC.....	49
Tabel 4.2 Hasil Pengujian 10 Siklus.....	52

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

Bab ini mengantarkan pembaca untuk dapat menjawab pertanyaan mengapa, apa yang diteliti, untuk apa suatu penelitian dilakukan. Jawaban pertanyaan tersebut akan diuraikan pada bab ini yang meliputi latar belakang, perumusan masalah, tujuan, metodologi, sistematika dan relevansi.

1.1 Latar Belakang

Umumnya penggunaan truk maupun bantuan tenaga manusia dalam pemindahan material memiliki kekurangan seperti keterlambatan pemindahan material, keterbatasan stamina manusia hingga kecelakaan kerja yang mengakibatkan terganggunya kegiatan industri manufaktur. Sistem *conveyor* memungkinkan transportasi material dengan lebih efisien dibandingkan menggunakan transportasi pengangkut seperti truk atau tenaga manusia. *Conveyor* merupakan peralatan yang digunakan dalam industri untuk memindahkan barang dari satu lokasi ke lokasi lainnya. Dengan adanya *conveyor*, pemindahan barang menjadi lebih mudah, cepat dan aman serta dapat menghemat waktu pengoperasian.

Proses pada *conveyor* tidak sekedar memindahkan barang saja tetapi juga terdapat proses lain ketika *conveyor* bekerja seperti menghitung jumlah barang, pengisian cairan, memisahkan atau menggabungkan produk, dan lain sebagainya. Tentunya proses-proses tersebut dapat berlangsung jika *conveyor* dilengkapi sensor dan aktuator tertentu. Dalam pengoperasiannya, serangkaian proses yang kompleks tersebut dapat diotomatiskan dengan menggunakan PLC sebagai kontroler. PLC bekerja berdasarkan instruksi yang diberikan pengguna dalam bentuk program, dimana bahasa pemrograman yang umum digunakan adalah *ladder diagram*.

Kompleksnya proses otomasi pada *conveyor* membutuhkan penggunaan *relay* yang banyak dalam pemrograman PLC. Kebutuhan *relay* tersebut relatif terbatas pada PLC yang rendah spesifikasinya. Tentunya pengadaan PLC dengan spesifikasi tinggi membutuhkan biaya yang tidak sedikit. Agar dapat mengatasi permasalahan tersebut maka diperlukan suatu metode yang dapat meminimalkan jumlah *relay* yang digunakan. Dalam pemrograman PLC, jumlah *relay* yang sedikit juga membuat program menjadi lebih singkat.

1.2 Perumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah bagaimana mengkonstruksi *ladder diagram* dengan menggunakan metode *Flow-Table/State Diagram* untuk proses otomasi *plant dual conveyor*.

1.3 Tujuan

Tujuan dari penelitian ini adalah mengkonstruksi *ladder diagram* dari proses otomasi *plant dual conveyor* dengan metode *Flow-Table/State Diagram*. Hasil konstruksi tersebut diharapkan dapat diimplementasikan pada proses kerja *plant dual conveyor*.

1.4 Metodologi

Kegiatan penelitian pada tugas akhir ini mencakup studi literatur yang dilakukan dengan membaca beberapa referensi mengenai topik yang akan diangkat sampai diperoleh gambaran yang jelas mengenai sistem yang akan dirancang. Pada penelitian yang dilakukan diketahui bahwa permasalahan pada sistem dapat diselesaikan dengan menggunakan metode *state diagram*. Permasalahan yang dihadapi adalah bagaimana meminimalkan jumlah *relay* pada program agar dapat memenuhi spesifikasi PLC yang relatif terbatas.

Dari permasalahan yang telah diketahui selanjutnya dipelajari mengenai metode yang dapat mengatasi permasalahan tersebut dan penggunaannya pada sistem. Metode *state diagram* dapat mengatasi banyaknya penggunaan *relay* pada sistem. Dimulai dengan membuat *state diagram* (I/O) dari sistem *dual conveyor* sesuai dengan kondisi sistem yang telah dideskripsikan. Informasi pada diagram tersebut disusun ke dalam *primitive flow table* dan disederhanakan menjadi *merged flow table* untuk mengetahui jumlah *relay* yang dibutuhkan sistem.

Merged flow table yang sudah didapatkan diubah menjadi gambar *state diagram* (R/O). *State* yang didapatkan menggambarkan jumlah *relay* yang digunakan. Setelah itu konstruksi *ladder diagram* sistem dapat dilakukan sesuai dengan informasi *state diagram* (R/O). Program yang telah dibuat kemudian disimulasikan terlebih dahulu dengan software PLC sebelum diimplementasikan ke *plant*.

1.5 Sistematika

Penulis membagi laporan penelitian ini menjadi lima bab yang terhubung satu sama lain. Hal ini untuk menghindari kesalahan interpretasi terhadap isi yang terdapat di dalam laporan. Penjelasan tentang masing-masing bab dibuat dengan sistematika penulisan sebagai berikut :

Bab I Pendahuluan

Bab ini meliputi latar belakang, perumusan masalah, tujuan, metodologi, sistematika dan relevansi.

Bab II Dasar Teori

Bab ini membahas tinjauan pustaka yang membantu penelitian, di antaranya adalah teori pemodelan *Flow-Table/State Diagram*, teori sistem *plant dual conveyor*, teori instrumentasi sistem tentang sensor dan aktuator yang digunakan serta teori otomasi sistem tentang pemrograman *ladder diagram* pada PLC .

Bab III Perancangan Sistem

Bab ini membahas perancangan sistem yang meliputi perancangan langkah sistem *plant dual conveyor*, pemodelan sistem otomasi *plant dual conveyor* dengan metode *Flow-Table/State Diagram*, perancangan *ladder diagram* sehingga membantu pembaca dalam memahami tahapan dari setiap proses dalam sistem yang dirancang.

Bab IV Hasil dan Analisa

Bab ini memuat hasil penerapan pemodelan yang telah dibuat dan analisisnya.

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Penyederhanaan jumlah *relay* menjadi hal yang sangat penting jika menggunakan PLC yang relatif kecil spesifikasinya. Tentunya hal tersebut berpengaruh terhadap biaya pengadaan suatu proyek di industri. Dengan menggunakan metode *state diagram* seperti pada penelitian ini, jumlah *relay* dapat diminimalkan. Hasil penelitian yang dilakukan oleh penulis diharapkan dapat dijadikan referensi untuk metode penyederhanaan jumlah *relay* pada *ladder diagram* dan dapat dikembangkan menjadi lebih baik kedepannya.

[Halaman ini sengaja dikosongkan]

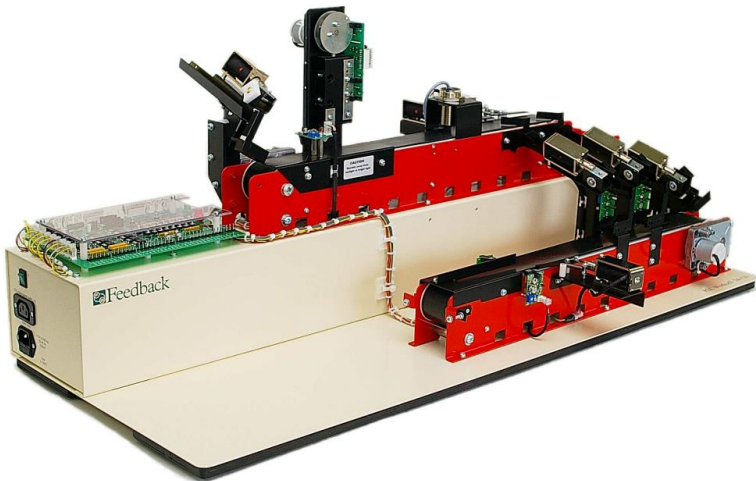
BAB 2

DASAR TEORI

Suatu teori diperlukan sebagai landasan maupun dasar untuk penulis dalam melakukan sebuah penelitian. Teori-teori tersebut dikaji terlebih dahulu untuk menunjang serta memperkuat penelitian penulis. Pada bab ini akan diuraikan teori mengenai *plant*, kontroler dan metode yang digunakan untuk mengatasi masalah pada tugas akhir ini.

2.1 *Dual Conveyor Workcell Systems*

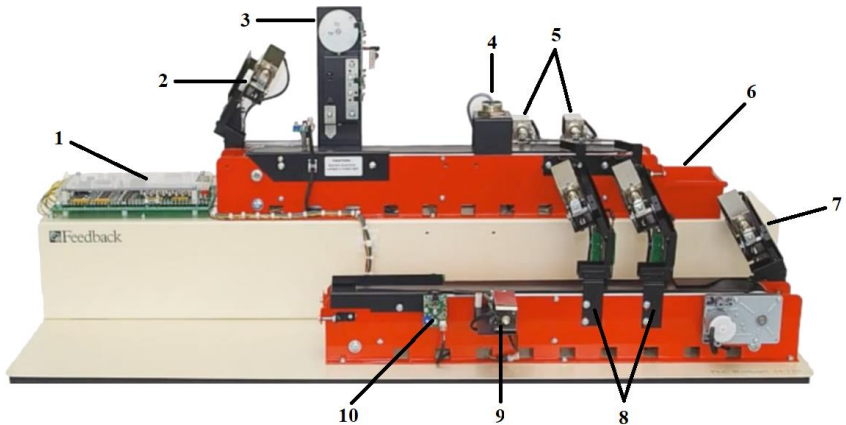
Plant yang digunakan oleh penulis dalam penelitian adalah sistem *dual conveyor*. Terdiri dari *conveyor* atas dan bawah, dengan berbagai sensor dan aktuator yang terhubung ke PLC seperti yang ditunjukkan oleh Gambar 2.1. Secara umum sistem ini memiliki fungsi untuk melakukan proses seleksi dan perakitan *part* atau benda kerja.



Gambar 2.1 *Dual Conveyor Workcell Systems*

Plant yang digunakan oleh penulis dalam penelitian adalah sistem *dual conveyor*. Terdiri dari *conveyor* atas dan bawah, dengan berbagai sensor dan aktuator yang terhubung ke PLC. Secara umum sistem ini memiliki fungsi untuk melakukan proses seleksi dan perakitan *part* atau

benda kerja. Proses seleksi dilakukan berdasarkan ketinggian yang telah ditentukan dan jenis materialnya (metal atau plastik). Proses perakitan dilakukan dengan merakit benda kerja yang memiliki pasak atau poros sebagai 'basis' dengan benda kerja yang disebut 'washer' atau berbentuk cincin yang memiliki lubang ditengahnya, dimana benda kerja yang dirakit telah terseleksi sesuai standar.



Gambar 2.2 Bagian *Dual Conveyor*

Gambar 2.2 menunjukkan bagian-bagian dari *dual conveyor* secara umum. Berikut ini adalah keterangan dari setiap bagian :

1. *Interface PCB*
2. *Parts Dispenser (Washer)*
3. *Height Gauge*
4. *Inductive Transducer*
5. *Parts Manipulators (Flippers)*
6. *Reject Box*
7. *Parts Dispenser (Peg)*
8. *Assembly Chutes*
9. *Flipper*
10. *Opto-sensor*

2.1.1 Power Supply dan Interface

Pada Gambar 2.3, catu daya untuk *conveyor*, sensor dan aktuator ditempatkan di bagian belakang pelat dasar, di bawah *conveyor* atas.

Interkoneksi antara PLC, sensor dan aktuator pada sistem membutuhkan rangkaian *interface*.



Gambar 2.3 Lokasi *Power Supply* dan *Interface PCB*

Rangkaian tersebut berada pada PCB yang terletak di sebelah kiri *conveyor* atas. Daya untuk PLC dihubungkan dengan menggunakan steker terpisah ke stopkontak utama yang terletak di sisi kiri.

2.1.2 Benda Kerja (*Part*)

Benda kerja (*part*) yang digunakan pada sistem *dual conveyor* terdiri dari dua jenis yaitu *part* dasar atau 'base' yang memiliki pasak atau poros (*peg*) dan *part* 'washer' yang memiliki lubang ditengahnya (lihat pada Gambar 2.4).



Gambar 2.4 *Part* untuk *assembly*

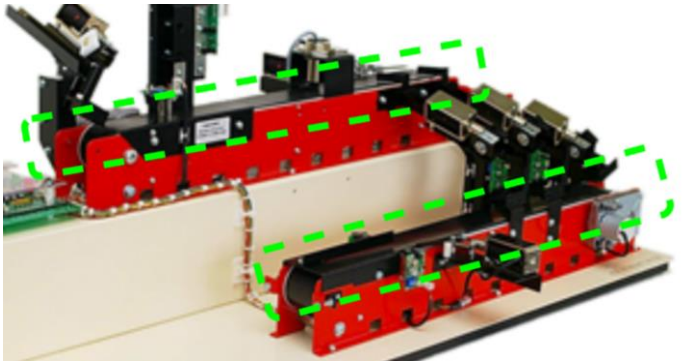
Material ‘washer’ dapat berupa bahan metal atau plastik dan memiliki ketinggian (ketebalan) yang berbeda. Ketinggian yang berbeda digunakan untuk membedakan benda kerja yang standar (benar) dan tidak standar (salah). Tabel 2.1 menunjukkan ukuran ketinggian dari benda kerja. Benda kerja dirakit dengan menghubungkan bagian pasak pada ‘base’ dengan lubang yang terdapat pada ‘washer’.

Tabel 2.1 Kriteria ketinggian benda kerja

<i>Part</i>	Tinggi (Ketebalan)
Standar, benar	8 mm
Tidak standar, salah	7 mm dan 9 mm

2.1.3 *Conveyor*

Conveyor merupakan alat transportasi yang digunakan dalam industri untuk memindahkan barang atau material dari suatu lokasi ke lokasi lainnya. Penggunaan truk maupun bantuan tenaga manusia untuk memindahkan material memiliki kekurangan mulai dari lambatnya pemindahan material, keterbatasan stamina manusia hingga kecelakaan kerja. Akibatnya kegiatan di industri menjadi tidak lancar.



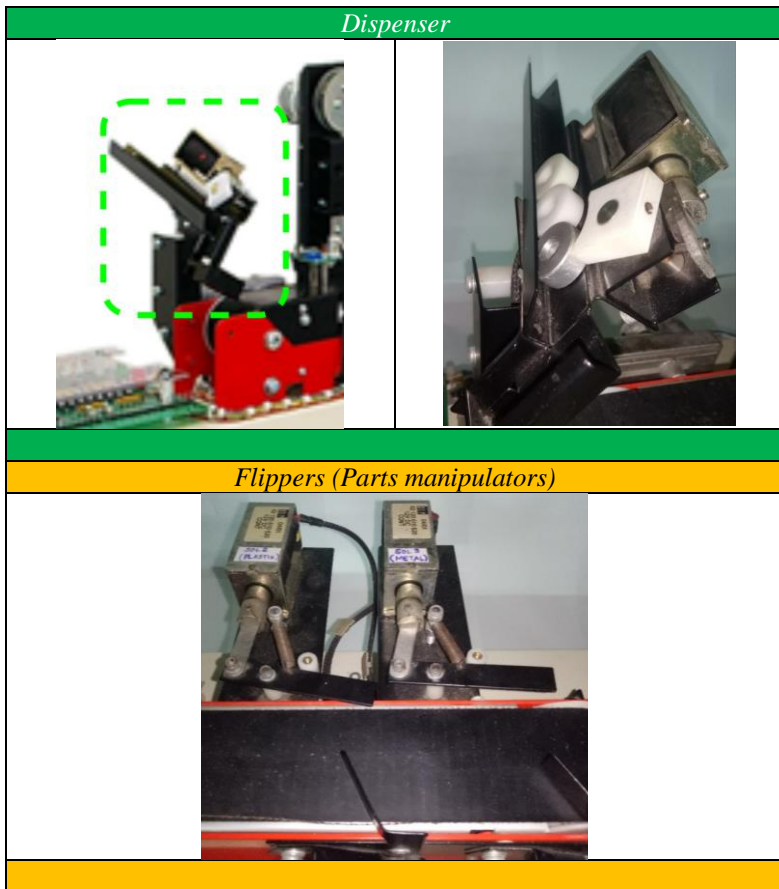
Gambar 2.5 *Dual Conveyor*

Kelebihan yang ada pada *conveyor* membuat pemindahan material menjadi lebih mudah, cepat dan aman serta dapat menghemat waktu pengoperasian. Sistem ini memiliki dua buah *conveyor* yang ditunjukkan

pada Gambar 2.5, digunakan untuk mengangkat benda kerja dan digerakkan oleh motor DC. Motor dikendalikan oleh PLC melalui rangkaian *interface*.

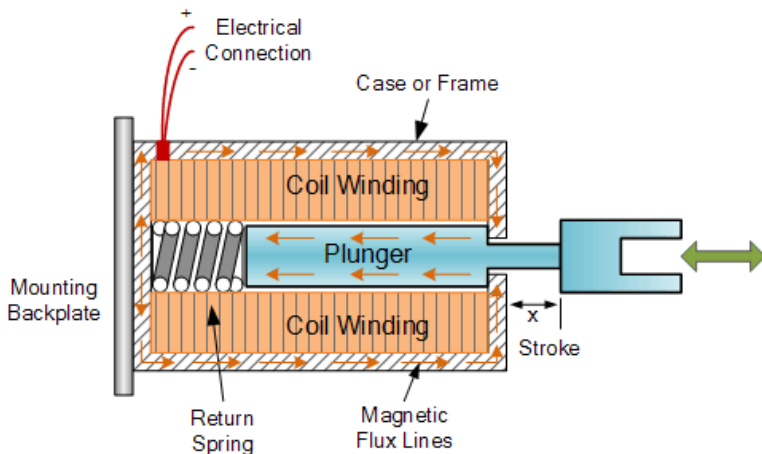
2.1.4 Dispenser dan Flipper

Dispenser dan *flipper* seperti pada Gambar 2.6 menggunakan aktuator *solenoid* dan telah dirancang untuk menangani dua jenis komponen yang digunakan dalam tugas perakitan.



Gambar 2.6 *Dispenser dan Flipper*

Dispenser berfungsi untuk mengeluarkan benda kerja dari tempat penyimpanan. Terdapat 4 buah *dispenser*, tiga untuk mengeluarkan *part* ‘base’ dan satu untuk mengeluarkan *part* ‘washer’. *Flipper* berfungsi untuk memindahkan *part* ‘standar’ dari *conveyor* atas ke *conveyor* bawah. *Flipper* akan dioperasikan oleh *solenoid* apabila *part* yang diseleksi ‘standar’. Saat *Flipper* aktif, *part* dipaksa bergerak ke tempat penyimpanan sementara sebelum dilakukan proses perakitan. *Flipper* tidak akan aktif apabila *part* yang diseleksi ‘tidak standar’ dan *part* akan dibawa ke tempat pembuangan.

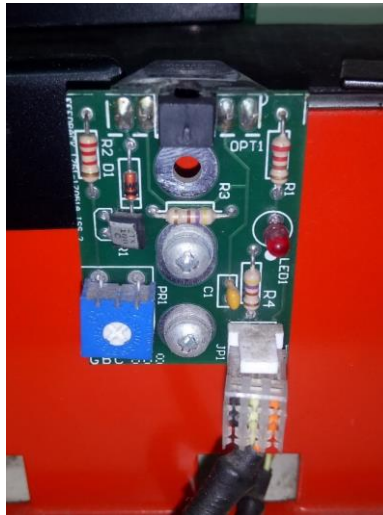


Gambar 2.7 Konstruksi *Solenoid* Linear tipe Tarik

Solenoid yang digunakan dalam sistem merupakan ‘*solenoid* linear tipe Tarik’ seperti pada Gambar 2.7. Dikatakan linear karena arah gerakan dan aksi dari ‘plunger’-nya (aktuator magnetik) dalam satu arah lurus. Konfigurasi pada *solenoid* linear tipe tarik (“Pull-Type”) adalah menarik beban yang terhubung ke arah dirinya sendiri saat diberi energi. Beban akan kembali ke posisi semula dengan bantuan pegas balik (*return spring*) saat tidak diberi energi.

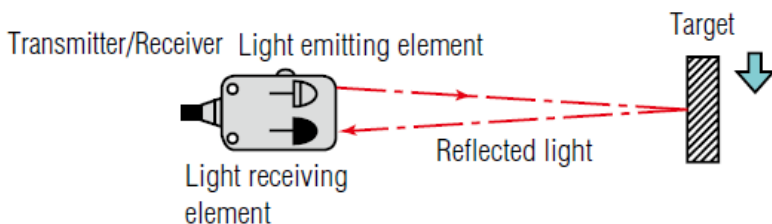
2.1.5 *Opto-detector*

Sebanyak 6 buah *opto-detector* dipasang di beberapa bagian pada sistem untuk mendeteksi ada atau tidaknya benda kerja. Gambar 2.8 menunjukkan *opto-detector* yang digunakan pada sistem.



Gambar 2.8 *Opto-detector*

Opto-detector merupakan jenis sensor fotoelektrik yang memanfaatkan sinar inframerah. Keadaan aktif atau tidaknya setiap *opto-detector* ditunjukkan oleh LED yang terpasang pada PCB detektor tersebut.



Gambar 2.9 Sensor Fotoelektrik Model Reflektif

Model sensor fotoelektrik yang ada pada *opto-detector* adalah model reflektif (lihat Gambar 2.9). Tipe reflektif digunakan untuk mendeteksi sinar yang dipantulkan dari target (dalam hal ini adalah benda kerja). Sinar dipancarkan dari elemen pemancar (*transmitter*) seperti LED dan diterima oleh elemen penerima (*receiver*) seperti *photodiode* maupun *phototransistor*.

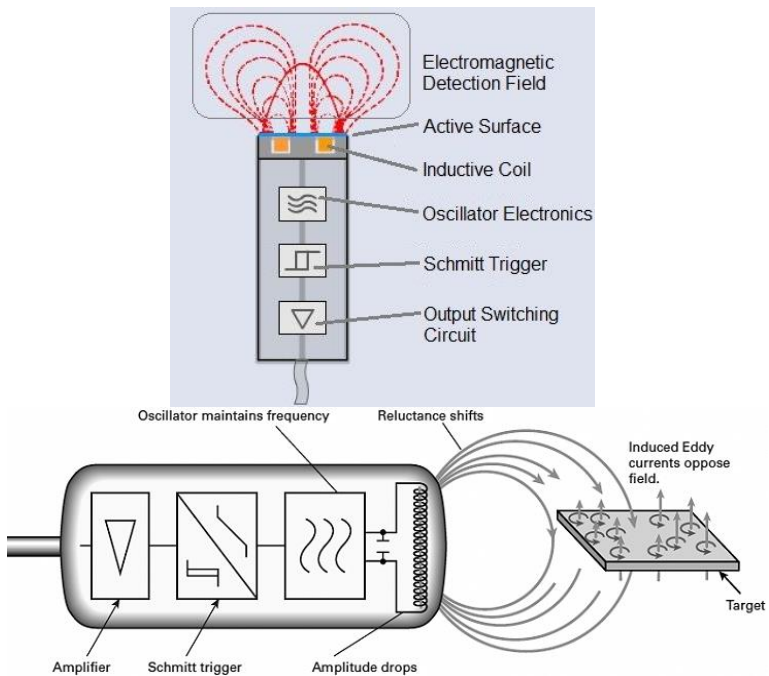
2.1.6 Sensor Induktif

Sistem juga dilengkapi dengan perangkat induktif yang digunakan bersamaan dengan sensor *opto* (ditunjukkan oleh Gambar 2.10) untuk menjadi detektor sederhana. Perangkat ini digunakan untuk mendeteksi perbedaan antara benda kerja metal dan plastik.



Gambar 2.10 Sensor Induktif

Osilasi menghasilkan medan elektromagnetik di depan sensor, karena kumparan terletak tepat di belakang “muka” sensor. Nama teknis dari muka sensor adalah permukaan aktif (*Active Surface*). Ketika sepotong logam konduktif masuk ke daerah yang dibatasi oleh medan elektromagnetik, beberapa energi osilasi ditransfer ke logam sebagai target. Energi yang ditransfer ini muncul sebagai arus listrik bersirkulasi kecil yang disebut dengan ‘arus eddy’ (*eddy current*). Inilah sebabnya mengapa sensor induktif terkadang disebut juga sensor arus eddy.



Gambar 2.11 Struktur Sensor Induktif

Gambar 2.11 menunjukkan struktur sensor induktif. Arus eddy yang mengalir mendapatkan perlawanan hambatan listrik saat mereka mencoba untuk bersirkulasi. Hal ini menciptakan sejumlah kecil daya hilang dalam bentuk panas (seperti pemanas listrik kecil). Kehilangan daya tidak sepenuhnya digantikan oleh sumber energi pada internal sensor, sehingga level atau intensitas dari amplitudo osilasi sensor menurun.

Level amplitudo yang menurun hingga di bawah ambang batas (*Threshold*) yang telah ditentukan, dideteksi oleh *Schmitt Trigger* (rangkain internal lainnya di dalam sensor). Ambang batas ini adalah level dimana kehadiran target logam dikonfirmasi. Setelah target terdeteksi oleh *Schmitt Trigger*, *output* pada sensor diaktifkan.

2.1.7 Height Detector

Detektor ini dapat menentukan tinggi (ketebalan) dari benda kerja dengan menggunakan *opto-detector* yang dapat diatur dan disesuaikan untuk menghasilkan *output* ketika benda kerja dengan ukuran yang ‘benar’ (8 mm) terdeteksi. Tampilan fisik detektor ketinggian dapat dilihat pada Gambar 2.12.



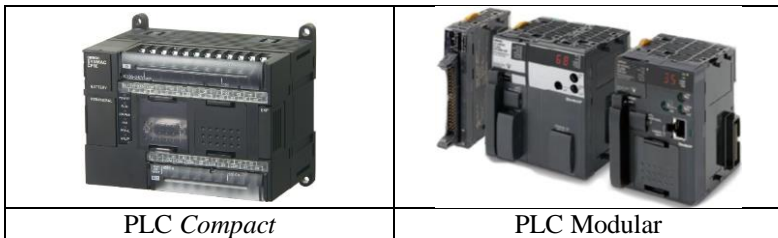
Gambar 2.12 Pengukur Ketinggian Part ‘Washer’

Benda kerja yang terlalu kecil (7 mm) dan terlalu besar (9 mm) akan ditolak. LED yang terpasang pada *height detector* bertindak sebagai indikator ukuran.

2.2 Programmable Logic Controller (PLC)

PLC merupakan sebuah perangkat elektronik yang dirancang untuk mengendalikan mesin dan proses di industri, dapat menyimpan instruksi dan menjalankan fungsi khusus seperti logika, urutan (*sequence*), pewaktuan (*timing*), penghitungan (*counting*) dan operasi aritmetika, serta tahan terhadap lingkungan industri yang keras.

Awal mula terbentuknya PLC adalah untuk menggantikan sistem kontrol berbasis *relay* (*relay logic control*) sebelum tahun 1960an. Salah satu kelemahan vital dari penggunaan susunan *relay* pada sistem kontrol adalah kesulitan dalam memodifikasi sistem yang sudah ada, baik karena ada kesalahan ataupun munculnya kebutuhan pengembangan sistem.



Gambar 2.13 PLC *Compact* dan Modular

Terdapat dua jenis PLC berdasarkan ukuran dan kemampuannya seperti yang ditunjukkan oleh Gambar 2.13, yaitu :

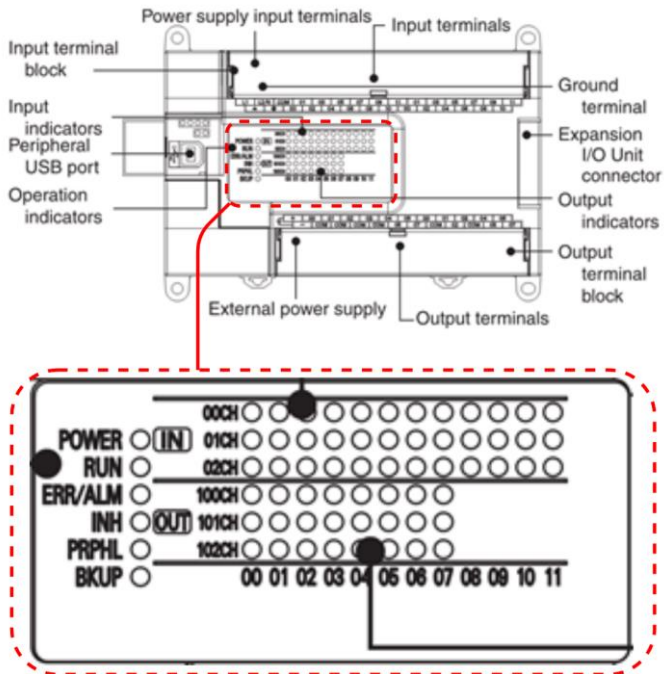
1. Tipe *Compact*

Tipe ini memiliki ciri-ciri berukuran kecil (*power supply*, CPU, modul *input/output* dan modul komunikasi menjadi satu), jumlah *input/output* yang relatif sedikit, terbatas dan tidak dapat ditambah (*expand*) serta tidak dapat ditambah modul-modul khusus.

2. Tipe Modular

Tipe ini berukuran besar, memungkinkan untuk ditambahkan modul *input/output* dan modul-modul khusus (*expandable*).

Pada penelitian ini, PLC digunakan sebagai kendali proses kerja dari *plant dual conveyor*. PLC yang digunakan pada penelitian ini adalah PLC Omron tipe *compact* dengan tipe perangkat CP1E dan tipe CPU E30. Memiliki 18 buah *input* digital dan 12 buah *output* digital.



Gambar 2.14 PLC Omron CP1E-30

Gambar 2.14 menunjukkan tampak atas dari PLC Omron CP1E-30 beserta indikatornya. Keterangan dari indikator ketika PLC beroperasi dijelaskan dalam Tabel 2.2.

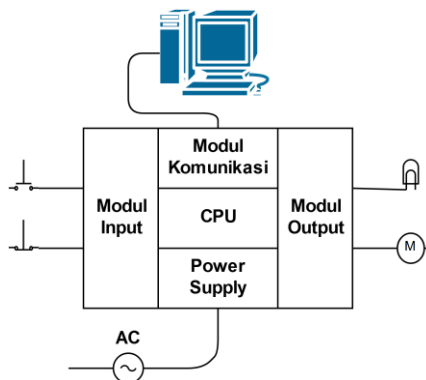
Tabel 2.2 Keterangan Indikator

No	Indikator	Status	Keterangan
1	POWER	<i>On</i>	PLC sedang terhubung dengan catu daya
		<i>Off</i>	PLC Tidak terhubung dengan catu daya
2	RUN	<i>On</i>	PLC beroperasi dalam mode MONITOR/RUN
		<i>Off</i>	PLC beroperasi dalam mode program atau terjadi kesalahan fatal

No	Indikator	Status	Keterangan
3	ERR/ALM	<i>On</i>	Terjadi kesalahan fatal
		<i>Blinking</i>	Terjadi kesalahan namun tidak fatal
		<i>Off</i>	PLC beroperasi normal
4	INH	<i>On</i>	Seluruh <i>output</i> berubah menjadi OFF
		<i>Off</i>	Kondisi <i>output</i> normal
5	PRPHL	<i>Blinking</i>	Sedang berjalan proses komunikasi data dengan komputer/laptop
		<i>Off</i>	Tidak ada komunikasi
6	BKUP	<i>On</i>	Program pengguna, parameter, dan kata-kata tertentu di area DM sedang ditulis ke memori cadangan (Riza, 2012).
		<i>Off</i>	Tidak ada proses <i>backup</i>
7	Input	<i>On</i>	Sinyal <i>input</i> dari <i>plant</i> aktif
		<i>Off</i>	Sinyal <i>input</i> dari <i>plant</i> nonaktif
8	Output	<i>On</i>	Sinyal <i>output</i> menuju <i>plant</i> aktif
		<i>Off</i>	Sinyal <i>output</i> menuju <i>plant</i> nonaktif

2.2.1 Komponen Dasar PLC

Dari Gambar 2.15 terlihat bahwa PLC terhubung dengan peralatan *input/output* (tombol, lampu, dsb.) dan PC (untuk kebutuhan pemrograman).



Gambar 2.15 Komponen Dasar PLC

Secara umum PLC tersusun oleh beberapa komponen diantaranya :

1. *Power supply*

Daya untuk PLC dapat berupa tegangan AC sebesar 120/240 VAC maupun tegangan DC sebesar 24 VDC. Selain itu PLC memiliki *power supply* internal (24 VDC) yang digunakan untuk menyediakan daya bagi peralatan I/O PLC.

2. *Prosesor (CPU)*

CPU bertugas untuk membaca, mengolah dan mengeksekusi instruksi program. CPU dapat mengerjakan tugas yang berhubungan dengan operasi logika dan aritmetika karena memiliki elemen kontrol ALU (*Arithmetic and Logic Unit*). Umumnya memori terletak di dalam CPU (satu modul) atau disebut memori internal. Apabila terdapat memori eksternal maka itu merupakan memori tambahan

3. *Modul Input-Output*

Bagian yang berfungsi sebagai perantara atau penghubung dari PLC ke peralatan *input* fisik (tombol, sensor, dsb.) dan *output* fisik (lampu, katup, dsb.) adalah modul *input-output*. Umumnya modul ini sudah terpasang secara internal di dalam PLC (ukuran *compact*). Untuk modul I/O yang terpisah dari CPU merupakan PLC modular.

4. *Modul Komunikasi*

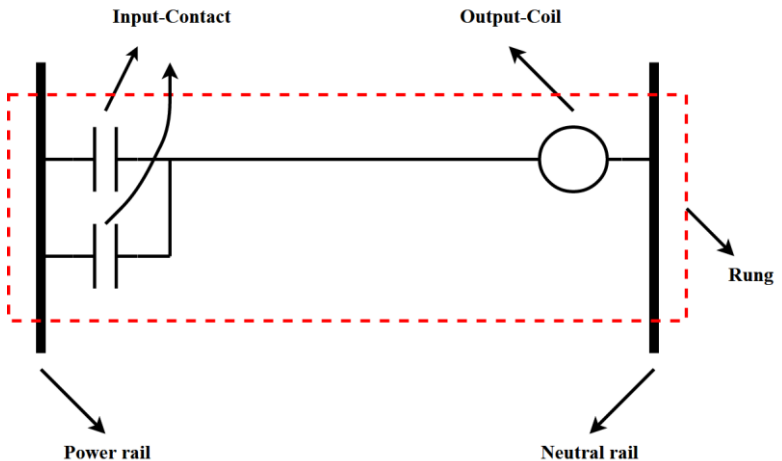
Koneksi antara CPU dan komputer (PC) diperlukan modul komunikasi agar dapat dilakukan pemrograman pada PLC. Selain itu juga untuk melakukan pemantauan (*monitoring*) maupun pertukaran data dengan perangkat lain.

2.2.2 Bahasa Pemrograman PLC

Berdasarkan standar yang telah ditentukan oleh IEC (*International Electrotechnical Commission*), badan standardisasi dunia dalam bidang teknik elektro, terdapat beberapa bahasa pemrograman PLC, yaitu :

1. *Ladder Diagram (LD)*

Merupakan bahasa pemrograman pertama yang diciptakan untuk PLC dan paling populer. Seperti namanya, Gambar 2.16 menunjukkan bahwa bentuk bahasa ini mirip dengan tangga (*ladder*).



Gambar 2.16 *Ladder Diagram*

Pada anak tangga (*rung*) terdapat komponen-komponen pemrograman LD seperti bagian *contact* (sebagai *input*) dan *coil* (sebagai *output*). *Rung* berada diantara dua garis vertikal, yaitu *power rail* dan *neutral rail* yang menggambarkan aliran program dari kiri ke kanan dan dari atas ke bawah seperti aliran arus listrik.

2. *Function Block Diagram (FBD)*

Istilah *function block diagram* pada pemrograman PLC adalah bahasa pemrograman dalam bentuk kotak atau blok grafis (lihat Gambar 2.17).



Gambar 2.17 Contoh Instruksi FBD

Block menggambarkan hubungan antara variabel *input* dan *output* yang memiliki fungsi khusus seperti instruksi dan logika aritmetika.

3. *Structured Text* (ST)

Bahasa pemrograman ini berbasis teks seperti bahasa pemrograman tingkat tinggi (PHP, Python dan C).

```
IF value < 7 THEN
  WHILE value < 8 DO
    value := value + 1;
  END_WHILE;
END_IF;
```

Gambar 2.18 Contoh Instruksi ST

Gambar 2.18 menunjukkan *syntax* dari ST dikembangkan mirip dengan *syntax* pada Bahasa pemrograman tingkat tinggi yang memiliki fungsi *loop*, variabel, kondisi dan operator.

4. *Instruction List* (IL)

Bahasa ini mirip dengan bahasa pemrograman tingkat rendah (Assembly) seperti pada Gambar 2.19.

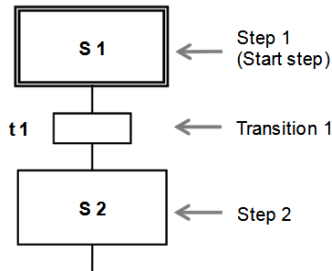
```
LD 2
MUL 2
ADD 3
ST Erg
```

Gambar 2.19 Contoh Instruksi IL

Sebuah IL terdiri dari sederetan instruksi dimana setiap instruksi dimulai dengan sebuah baris baru dan mengandung sebuah operator.

5. *Sequential Function Chart (SFC)*

SFC adalah bahasa berorientasi grafis yang menampilkan alur proses (lihat Gambar 2.20).

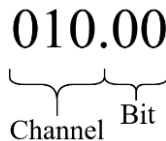


Gambar 2.20 Contoh Instruksi SFC

Bahasa ini memungkinkan pengguna untuk menggambarkan urutan kronologis berbagai tindakan dalam sebuah program.

2.2.3 Pengalaman dan Instruksi Dasar PLC

Alamat atau *addressing* pada PLC merupakan identitas dari komponen (*contact*, *coil*, dsb.) *ladder diagram* seperti halnya nama manusia. Masing-masing vendor PLC memiliki aturan pengalaman tersendiri. Berikut ini adalah aturan pengalaman yang digunakan pada PLC Omron.



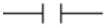



Gambar 2.21 Aturan Pengalaman PLC Omron

Penulisan alamat pada PLC Omron diawali dengan nomor *channel* (CH) dan diikuti dengan nomor *bit* dimana keduanya dipisah oleh tanda titik (.). Contoh pada Gambar 2.21 menunjukkan aturan *addressing* PLC

Omron di “10.00” yang berarti alamat berada di *channel* 10 dan *bit* ‘00’. Umumnya penentuan alamat pada *input* dan *output* PLC dapat dilihat pada *cover* PLC itu sendiri. Untuk *input* menggunakan CH0 dan CH1, untuk *output* menggunakan CH10, CH11, CH100 dan CH101 (bergantung pada tipe PLC).

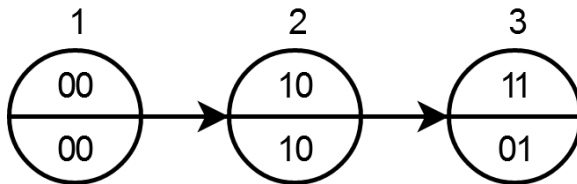
Instruksi atau perintah dasar pada PLC dirancang melakukan pemrograman pada PLC. Perintah-perintah tersebut dibutuhkan untuk menangani sistem kontrol otomatis pada mesin industri. Tabel 2.3 menampilkan beberapa instruksi dasar.

Tabel 2.3 Instruksi Dasar PLC

Instruksi	Simbol	Keterangan
<i>Load</i>		Instruksi <i>load</i> adalah kontak <i>normally open</i> (NO). Perintah ini digunakan untuk mengetahui kehadiran sinyal <i>input</i> fisik. Ketika <i>input</i> fisik aktif maka simbol juga akan aktif.
<i>Load Bar</i>		Instruksi <i>load bar</i> adalah kontak <i>normally closed</i> (NC). Kebalikan dari perintah <i>load</i> , ketika <i>input</i> fisik nonaktif maka simbol <i>load bar</i> akan aktif.
<i>Out</i>		Mirip seperti <i>coil</i> pada <i>relay</i> . Simbol akan aktif ketika mendapatkan sinyal benar (<i>True</i>) dari <i>rung</i> di sebelah kiri <i>ladder</i> .
<i>Out Bar</i>		Simbol akan aktif ketika mendapatkan sinyal salah (<i>False</i>) dari <i>rung</i> di sebelah kiri <i>ladder</i> . Oleh karena itu instruksi ini dapat dikatakan sebagai <i>output normally closed</i> atau kebalikan dari perintah <i>out</i> .

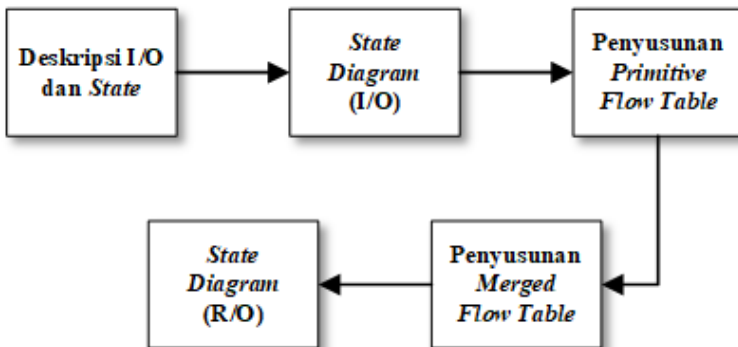
2.3 Flow-Table/State Diagram

Metode *flow-table* atau disebut juga *state diagram* merupakan suatu grafik yang merepresentasikan kejadian atau keadaan suatu sistem dalam bentuk lingkaran. Lingkaran yang dimaksud selanjutnya disebut dengan istilah *state*. Pada *state* tersebut berisi informasi *input* dan *output* dari sebuah sistem yang ditulis dalam digit biner seperti pada Gambar 2.22.



Gambar 2.22 *State diagram*

Gambar 2.23 menunjukkan langkah-langkah dalam membuat *state diagram* dimulai dengan mendeskripsikan *input/output* dan kondisi *input/output* dari setiap *state* untuk memudahkan perancangan.



Gambar 2.23 Langkah desain *state diagram*

Setelah itu dilakukan penyusunan *state diagram* (I/O) sesuai informasi yang ada dalam tabel. Dilanjutkan dengan membuat *primitive* dan *merged flow table*. Kemudian menyusun *state diagram* (R/O) sesuai dengan penyederhanaan yang dilakukan pada *merged flow table* hingga dikonversi ke dalam *ladder diagram*.

2.3.1 Deskripsi I/O dan State

Sebelum menggambar *state diagram*, keterangan dari setiap bit *input* dan *output* dituliskan terlebih dahulu (lihat pada Tabel 2.4).

Tabel 2.4 Keterangan I/O

Bit		INPUT		OUTPUT	
		Keterangan	Simbol	Keterangan	Simbol
i	j	(Contoh)	x_i	(Contoh)	z_j
$i+1$	$j+1$	(Sensor)	x_{i+1}	(Aktuator)	z_{j+1}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
m	n	(.....)	x_m	(.....)	z_n

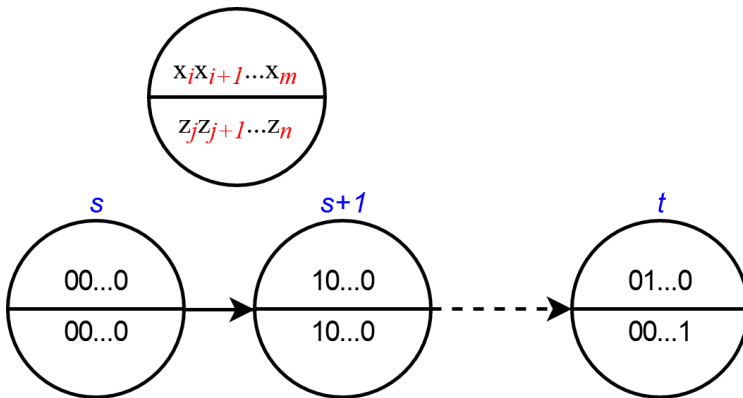
Selain itu, kejadian suatu sistem juga diterjemahkan ke dalam tabel (lihat Tabel 2.5). Tujuan pembuatan tabel adalah untuk memudahkan penulis dalam menyusun *state diagram* (I/O) di langkah berikutnya dan juga untuk mengetahui informasi yang terkandung dalam setiap *state*.

Tabel 2.5 Terjemahan urutan *state diagram* (I/O)

State	Kondisi Input	Kondisi Output
s	(Misal : Tombol Start ditekan)	(Conveyor aktif)
$s+1$	(Sensor opto aktif)	(Conveyor nonaktif)
\vdots	\vdots	\vdots
t	(.....)	(.....)

2.3.2 Penyusunan State Diagram (I/O)

Dalam penulisannya, semua *input/output* dalam digit biner dituliskan dalam sebuah *state* diurutkan mulai dari yang pertama hingga yang terakhir seperti menuliskan bilangan pecahan dengan susunan *input* per *output* (I/O).. Sebuah *input/output* dikatakan aktif apabila dinyatakan dalam digit biner '1', sebaliknya dikatakan nonaktif apabila dinyatakan dalam digit biner '0'. Banyaknya jumlah *input* ditulis dengan simbol " x_i " dan *output* dengan simbol " z_j " serta penomoran *state* ditulis dengan simbol " s " seperti yang ditunjukkan oleh Gambar 2.24. Anak panah menunjukkan transisi dari *state* sebelumnya ke *state* berikutnya. Sebuah *state* akan beralih ke *state* selanjutnya apabila terjadi perubahan pada bit *input* yang dapat berakibat berubahnya bit *output*.



Gambar 2.24 State Diagram (I/O)

2.3.3 Primitive Flow Table

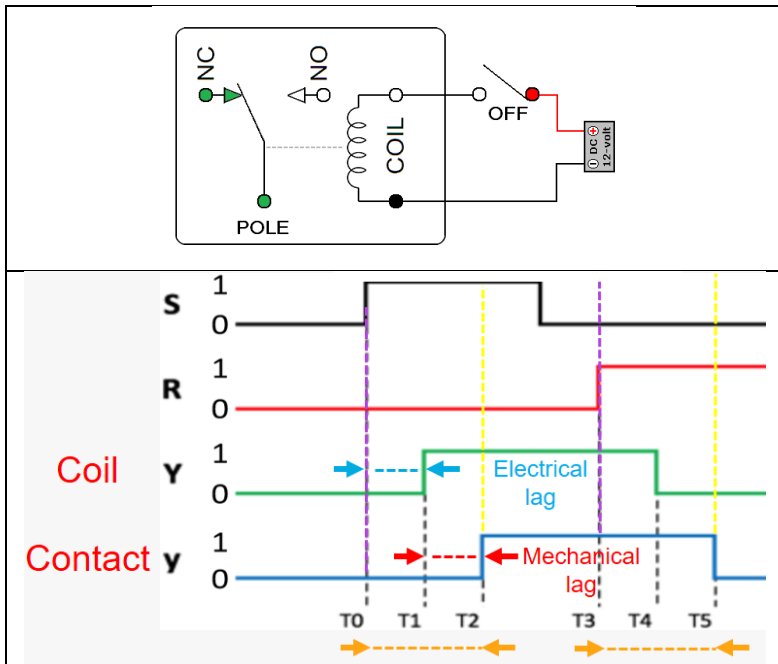
Penyusunan *state diagram* (I/O) yang telah selesai ditindaklanjuti dengan membuat *primitive flow table*. Tabel ini berisi informasi berupa posisi *state* stabil, *state* tidak stabil, dan *don't care* pada bit *input*, serta bit *output* pada *state* ke- s . Dimulai dengan mengisi kolom *Row*, *Inputs* $x_i x_{i+1} \dots x_m$, dan *Outputs* $z_j z_{j+1} \dots z_n$ sesuai jumlah bit *input/output* yang ada di *state diagram* (I/O). Kolom '*Row*' yang merupakan kolom untuk nomor *state* mulai dari *state* pertama " s " hingga *state* terakhir " t ".

Tabel 2.6 Kombinasi bit *input*

Primitive Flow Table								
Row	Inputs $x_i x_{i+1} \dots x_m$				Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	...	01...0	z_j	z_{j+1}	...	z_n
s								
$s+1$								
\vdots								
t								

Kolom "*Inputs*" berisi kombinasi bit *input* yang berbeda dari setiap *state*. Kombinasi bit *input* dituliskan mulai dari bit *input* pertama hingga bit terakhir di dalam 1 sel pada Tabel 2.6 (misal : 000, 100, dst). Jika terdapat kombinasi bit *input* yang sama dari *state* yang berbeda maka cukup ditulis 1 (satu) kali saja. Misalkan terdapat kombinasi bit *input* '100' pada *state* ke-1 dan *state* ke-5, maka tidak perlu ditulis 2 (dua) kali

dalam 2 sel namun cukup ditulis satu kali saja dalam 1 sel. Setelah kombinasi bit *input* selesai disusun, diteruskan dengan menyusun *state* stabil, *state* tidak stabil, dan *don't care* secara berurutan.



Gambar 2.25 Ilustrasi *State* Tidak Stabil dan *State* Stabil

Untuk memahami istilah *state* tidak stabil dan *state* stabil maka perhatikan respon *flip-flop* yang digambarkan dalam *sequence chart* (lihat Gambar 2.25). Ketika diberikan sinyal **SET** pada saat t_0 (garis vertikal warna ungu), sinyal *output* 'y' tidak langsung menuju kondisi HIGH seperti sinyal *input* 'SET' melainkan terdapat jeda (*lag*). Jeda yang dimaksud adalah saat t_0 hingga t_2 yang terdiri dari jeda elektrik (t_0-t_1) dan jeda mekanik (t_1-t_2). Jeda disebabkan oleh adanya induktansi pada *relay*, dimana jeda elektrik terjadi ketika *coil* (kumparan) belum teraliri sepenuhnya oleh arus. Arus teraliri sepenuhnya saat t_1 dan timbul gaya elektromagnet untuk menarik *contact*. Jeda mekanik terjadi saat *contact*

mulai bergerak (saat t_1) untuk menutup. *Contact* tertutup sepenuhnya saat t_2 .

Hal yang sama (jeda) juga terjadi ketika diberikan sinyal RESET pada saat t_3 . Arus kumparan sepenuhnya terputus saat t_4 dan *contact* akan terbuka sepenuhnya hanya pada saat t_5 . Jadi interval waktu ini (t_0 ke t_2 dan t_3 ke t_5) merepresentasikan “**state tidak stabil**”, yang artinya adalah keadaan saat *output* tidak konsisten atau tidak sama dengan sinyal *input*. Keadaan saat t_2 dan t_5 dimana sinyal *input* sama dengan *output* disebut dengan “**state stabil**”.

Tabel 2.7 *State stabil*

Primitive Flow Table								
Row	Inputs $x_i x_{i+1} \dots x_m$				Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	$s+I$						
$s+I$		$s+I$						
\vdots	\vdots	\vdots	\ddots	\vdots				
t				t				

State stabil (*state* ke- s) merupakan posisi *state* yang ditulis pada kolom tersebut dan memiliki kombinasi bit *input* yang sama dengan kombinasi bit *input* di *state diagram* (I/O). *State stabil* ditandai dengan penomoran yang ditulis tebal (di-**bold**). Semua *state stabil* yang ada di *state diagram* (I/O) harus diisi pada tabel Tabel 2.7.

Tabel 2.8 *State tidak stabil*

Primitive Flow Table								
Row	Inputs $x_i x_{i+1} \dots x_m$				Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	$s+I$						
$s+I$		$s+I$						
\vdots	\vdots	\vdots	\ddots	\vdots				
t				t				

State tidak stabil adalah representasi dari tanda panah atau transisi di *state diagram* (I/O). Tanda panah yang dimaksud adalah tanda panah yang keluar dari sebuah *state* dan menuju ke *state* berikutnya. Misalkan terdapat tanda panah yang keluar dari *state* 1 menuju ke *state* 2, maka tanda panah tersebut merupakan “*state tidak stabil* 2”, yang artinya *state* 1 dapat menuju ke *state* 2 dengan melalui “*state tidak stabil* 2”. Pada tabel

ditulis dengan cara memberi nomor yang sama dengan *state* tujuan tetapi **tidak** ditulis tebal. Penempatan *state* tidak stabil pada Tabel 2.8 berada pada kolom yang sama dengan kolom *state* tujuan tetapi masih dalam satu baris dengan *state* asal. Sama seperti *state* stabil di langkah sebelumnya, *state* tidak stabil yang terdapat pada *state diagram* (I/O) harus diisi semua pada tabel.

Tabel 2.9 *Don't care*

Primitive Flow Table								
Row	Inputs $x_i x_{i+1} \dots x_m$				Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	$s+1$...	-				
$s+1$	-	$s+1$...	-				
\vdots	\vdots	\vdots	\ddots	\vdots				
t	-	-	...	t				

Kondisi *don't care* adalah kondisi yang tidak berpengaruh pada penyusunan *state diagram* sehingga dapat diabaikan. Semua sel-sel pada Tabel 2.9 yang kosong setelah pengisian *state* stabil dan *state* tidak stabil, diisi dengan tanda “-“ untuk menyatakan kondisi *don't care*.

Tabel 2.10 Kombinasi bit *output*

Primitive Flow Table								
Row	Inputs $x_i x_{i+1} \dots x_m$				Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	s	...	-	0	0	...	0
$s+1$	-	$s+1$...	-			...	
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
t	-	-	...	t	0	0	...	1

Pada Tabel 2.10, kolom “*Outputs*” diisi dengan cara memasukkan kombinasi bit *output* dari setiap *state* ke dalam kolom masing-masing bit *output*. Misalkan terdapat kombinasi bit *output* ‘001’ pada *state* ke- s , maka bit *output* pertama (bit ‘0’) diisikan pada kolom “ z_j ”, bit *output* kedua (bit ‘0’) diisikan pada kolom “ z_{j+1} ”, dan bit *output* terakhir (bit ‘1’) diisikan pada kolom “ z_n ” secara berurutan. Secara umum Tabel 2.11 menunjukkan penyusunan *primitive flow table* berdasarkan *state diagram* (I/O) pada Gambar 2.24.

Tabel 2.11 *Primitive Flow Table*

Primitive Flow Table									
Row	Inputs $x_i x_{i+1} \dots x_m$					Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	11...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	s	-	...	-	0	0	...	0
$s+1$	-	$s+1$	$s+2$...	-	1	0	...	0
$s+2$	s	-	$s+2$...	-	0	0	...	0
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
t	-	-	-	...	t	0	0	...	1

2.3.4 Merged Flow Table

Merged flow table adalah tabel untuk menyederhanakan *output* dan menentukan jumlah *relay*. Informasi yang terkandung didalamnya akan digunakan untuk menyusun *state diagram* (R/O) dan konversi ke *ladder diagram*. Langkah pertama dalam menyusun *merged flow table* adalah menggabungkan baris. Baris yang akan digabung **disarankan** (tidak wajib) untuk memilih baris yang memiliki kombinasi bit *output* yang sama pada *primitive flow table*. Baris yang akan digabung (*merging*) harus memenuhi aturan berikut, antara lain :

1. Jika terdapat *state* stabil dan *state* tidak stabil pada kolom yang sama, baris yang digabungkan mendapatkan *state* stabil.
2. Jika terdapat nomor dan tanda “-” pada kolom yang sama, baris yang digabungkan mendapatkan nomor (dengan atau tanpa tulisan tebal, sesuai kasus yang mungkin terjadi).
3. Jika hanya terdapat tanda “-” pada kolom, baris yang digabungkan mendapatkan tanda “-”.

Tabel 2.12 Kombinasi bit *output* yang sama

Primitive Flow Table									
Row	Inputs $x_i x_{i+1} \dots x_m$					Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	11...0	...	01...0	z_j	z_{j+1}	...	z_n
s	s	s	-	...	-	0	0	...	0
$s+1$	-	$s+1$	$s+2$...	-	1	0	...	0
$s+2$	s	-	$s+2$...	-	0	0	...	0
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
t	-	-	-	...	t	0	0	...	1

Pada Tabel 2.12 terdapat baris yang *output*-nya sama meskipun *state*-nya berbeda, yaitu baris “ s ” dan “ $s+2$ ”. Baris-baris tersebut dapat

digabungkan jika memenuhi aturan yang telah disebutkan sebelumnya menjadi Tabel 2.13.

Tabel 2.13 Merged Flow Table (1)

Merged Flow Table									
Row	Inputs $x_i x_{i+1} \dots x_m$					Outputs $z_j z_{j+1} \dots z_n$			
	00...0	10...0	11...0	...	01...0	z_j	z_{j+1}	...	z_n
$s, s+2$	s	s	s+2	...	-	0	0	...	0
$s+1$	-	s+1	s+2	...	-	1	0	...	0
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots
t	-	-	-	...	t	0	0	...	1

Penggabungan baris sebenarnya bertujuan untuk meminimalisasi jumlah *relay*. Terdapat beberapa alasan dalam meminimalkan *relay*, diantaranya :

1. Sebuah program yang panjang membutuhkan waktu lebih lama untuk masuk ke dalam memori PLC.
2. Sebuah program yang panjang umumnya lebih sulit untuk dipahami atau dicek.
3. Jika program terlalu panjang, kapasitas memori mungkin tidak cukup bila menggunakan PLC yang murah dan kecil (alasan ekonomi).
4. Tingkat pemindaian PLC tergantung pada jumlah memori yang digunakan. Dengan demikian semakin pendek program, semakin cepat PLC merespon.

Setelah menggabungkan baris, langkah berikutnya adalah menentukan jumlah *relay* yang akan digunakan. Banyaknya jumlah *relay* yang akan digunakan harus dapat meng-cover jumlah baris atau *state* yang telah digabung. Hal ini terkait dengan teori rangkaian RS flip-flop yang dapat menghasilkan dua *state*, yaitu saat *output* bernilai '0' atau '1'. Oleh karena itu, n buah flip flop atau *relay* dapat membedakan sebanyak 2^n *state* yang berbeda.

Jika terdapat 3 *state* hasil dari penggabungan baris, maka dibutuhkan minimal 2 (dua) buah flip-flop sehingga menghasilkan 4 (dari 2^2) kombinasi bit yang berbeda untuk membedakan 4 *state* meskipun terdapat 1 *state* yang tidak digunakan. Total ada 4 baris hasil dari penggabungan sehingga diperlukan minimal 2 buah *relay* (y_1 dan y_2) yang menghasilkan 4 (dari 2^2) kombinasi bit yang berbeda, diantaranya 00, 01, 11, dan 10.

Penyusunan kombinasi bit pada kolom “*System’s*” hanya diperbolehkan terjadi perubahan bit sebanyak satu bit saja (misal : 01 ke 11) mengikuti perubahan *state* asal ke *state* tujuan.

Tabel 2.14 *Merged Flow Table (2)*

Merged Flow Table									
Row	Inputs x_1x_2				Outputs $z_1z_2z_3$			System’s	
	10	00	01	11	z_1	z_2	z_3	y_1	y_2
1, 2	1	2	3	-	0	0	0	0	0
3, 4	5	4	3	-	1	0	0	1	0
5, 6	5	6	7	-	1	1	0	1	1
7, 8	1	8	7	-	0	1	0	0	1

Ilustrasi *merged flow table* yang sudah selesai disusun ditunjukkan oleh Tabel 2.14. Tanda panah pada Tabel 2.15 menunjukkan arah perpindahan *state*.

Tabel 2.15 *Merged Flow Table (3)*

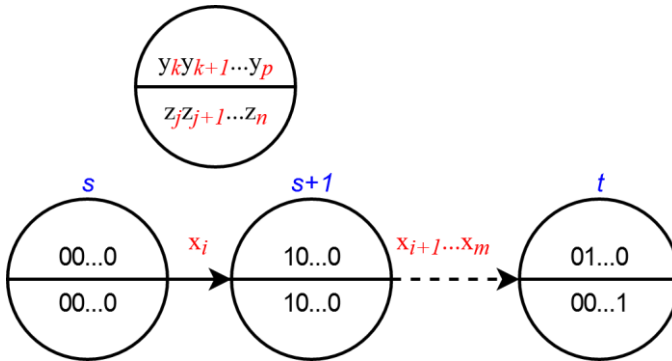
Merged Flow Table									
Row	Inputs x_1x_2				Outputs $z_1z_2z_3$			System’s	
	10	00	01	11	z_1	z_2	z_3	y_1	y_2
1, 2	1	2	3	-	0	0	0	0	0
3, 4	5	4	3	-	1	0	0	1	0
5, 6	5	6	7	-	1	1	0	1	1
7, 8	1	8	7	-	0	1	0	0	1

2.3.5 Penyusunan *State Diagram (R/O)*

Langkah terakhir dari metode *state diagram* adalah menyusun *state diagram* (R/O) dengan menggunakan informasi pada *merged flow table*. Untuk penulisannya sama seperti *state diagram* (I/O), hanya saja isi dari setiap *state* bukan dalam format *input/output* melainkan dalam format *relay/output*.

Pada baris 1 yang baru (lihat pada Tabel 2.15), hasil penggabungan antara baris 1 dan 2 diketahui bit *relay* ‘00’ dan bit *output* ‘000’. Nantinya format penulisan di dalam “*state* 1 baru” menjadi “00/000”. Pada transisi (tanda panah) diberi informasi bit *input* apa yang mengubah *state* sebelumnya ke *state* selanjutnya seperti “ x_i ” dan seterusnya.. Informasi

pada *merged flow table* dipindahkan semua ke dalam *state diagram* (R/O) secara lengkap. Ilustrasi penulisan dari *state diagram* (R/O) ditunjukkan oleh Gambar 2.26.



Gambar 2.26 State Diagram (R/O)

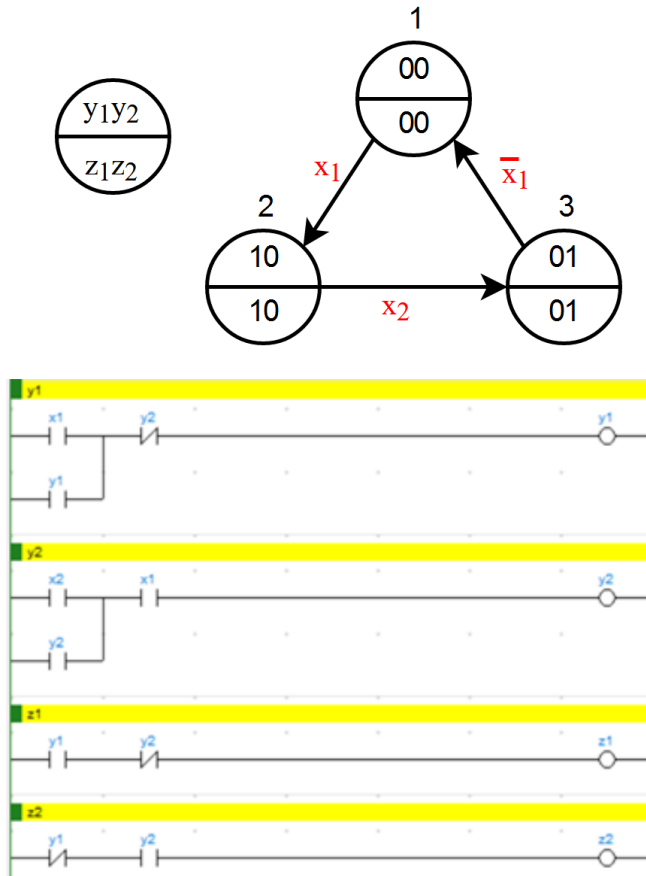
2.4 Konstruksi State Diagram (R/O)-Ladder

Agar dapat melakukan pemrograman melalui PLC maka *state diagram* (R/O) yang telah disusun harus diterjemahkan ke dalam *ladder diagram*. Pada Gambar 2.26, terdapat dua buah bit *relay* (“*y*₁” dan “*y*₂”) dan dua bit *output* (“*z*₁” dan “*z*₂”). Caranya dimulai dengan mengkonstruksi bagian *ladder* dari setiap *relay* dan dilanjutkan dengan bagian *ladder* pada setiap *output*.

Untuk penyusunan *ladder* dari *relay*, terlebih dahulu harus mengetahui perubahan bit *relay* pada setiap *state*. Bit *relay* dikatakan aktif apabila bit bernilai ‘1’ dan dikatakan nonaktif apabila bit bernilai ‘0’. Berubahnya bit *relay* dari bit ‘0’ (di *state* asal) ke bit ‘1’ (di *state* tujuan) yang disebabkan adanya suatu *input* disebut dengan *set*. Sebaliknya jika bit *relay* berubah dari bit ‘1’ ke bit ‘0’ disebut dengan *reset*.

Perhatikan *state* (R/O) pada Gambar 2.27, *relay* “*y*₁” mengalami perubahan dari nonaktif (pada *state* 1) menjadi aktif (pada *state* 2) oleh *input* “*x*₁”. Kemudian *relay* “*y*₁” menjadi nonaktif ketika ada *input* “*x*₂”. Jika ingin membuat *ladder* untuk *relay* “*y*₁” maka “*y*₁” menjadi *output* (*coil*) pada *rung* tersebut dan dicari *set* dan *reset*-nya. Yang menjadi *set* untuk *relay* “*y*₁” adalah *input* “*x*₁” dan *relay* “*y*₁” itu sendiri (*self-holding*).

“ x_1 ” dan “ y_1 ” ditulis dalam bentuk NO (*Normally Open*). Jika dituliskan, *set* untuk *relay* “ y_1 ” adalah “ x_1 ” di-*or* (paralel) dengan “ y_1 ”.



Gambar 2.27 State Diagram (R/O) ke Ladder

Pada terjemahan *ladder* untuk *reset*, tidak ada ketentuan khusus sehingga dapat ditentukan sendiri tergantung urutan kejadian pada sistem. Misalkan *output* “ z_1 ” akan nonaktif saat *output* “ z_2 ” aktif. Sedangkan *output* “ z_2 ” akan aktif ketika *relay* “ y_2 ” aktif. Dari sini *programmer* dapat

memilih apakah ingin menggunakan “z₂” atau “y₂” sebagai *reset* pada *relay* “y₁”.

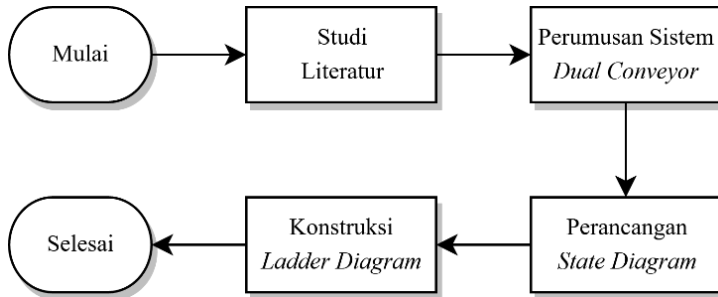
Pada contoh di Gambar 2.27, penulis memilih “y₂” sebagai *reset* untuk *relay* “y₁” sehingga “y₂” ditulis dalam bentuk NC dan diletakkan sebelum *output* (*coil*) pada *rung* program. Didapatkan susunan *set* dan *reset* dari *relay* “y₁” adalah “(x₁ + y₁) . y₂”. Sama seperti *relay* “y₁”, untuk *relay* “y₂” juga dicari dahulu yang menjadi *set* dan *reset*-nya dan setelah itu diterjemahkan ke *ladder*.

Bagian *ladder* untuk *output* cukup dengan mengetahui kombinasi bit *relay* dimana bit *output* tersebut aktif. Misalkan pada *state* 2, bit *output* “z₁” aktif (bernilai biner ‘1’) dengan kombinasi bit *relay*-nya ‘10’ maka bit ‘10’ tersebut diubah menjadi $y_1 \cdot \overline{y_2}$. Untuk itu penulisan *ladder* dari *output* “z₁” adalah dengan menuliskan “y₁ . $\overline{y_2}$ ” menjadi *input* dan “z₁” ditulis menjadi *coil* pada *rung* program.

BAB 3

PERANCANGAN SISTEM

Tahapan penelitian yang dilakukan dalam menjawab perumusan masalah terdiri dari beberapa bagian. Dimulai dari studi literatur, perumusan sistem *dual conveyor*, perancangan *state diagram*, dan dilanjutkan dengan penyusunan *ladder diagram* seperti yang ditunjukkan Gambar 3.1.



Gambar 3.1 Tahapan penelitian

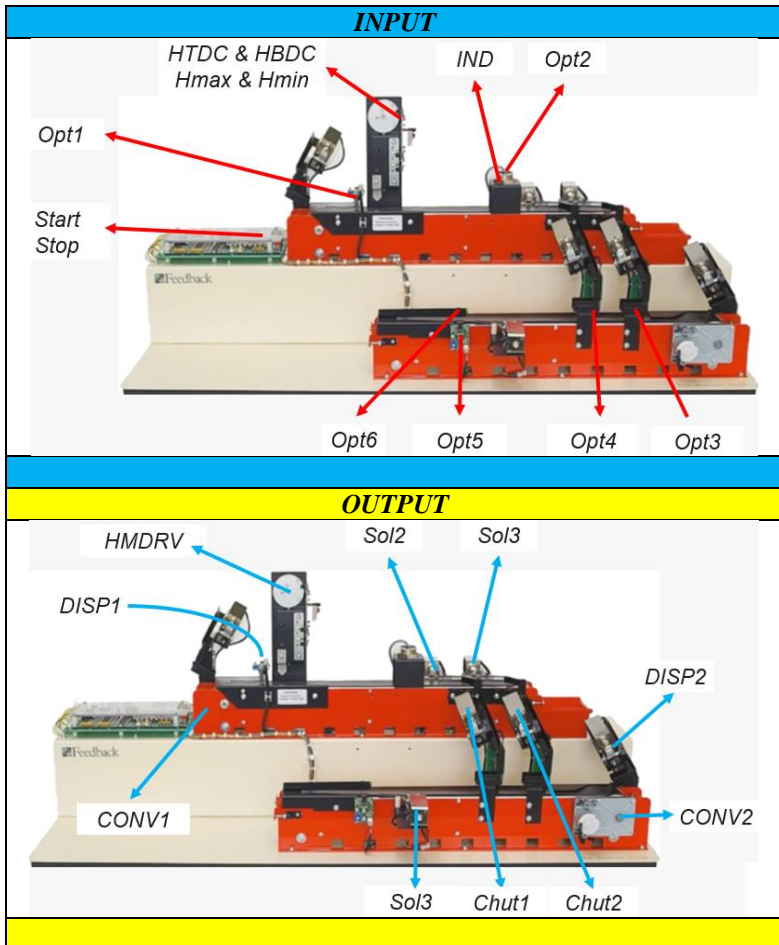
Pada tahapan studi literatur, penulis mempelajari tentang *plant* dan metode yang digunakan dalam penelitian. Tahap selanjutnya dilakukan perumusan sistem dari *plant dual conveyor* dengan mempelajari bagian *input/output* sistem sehingga dapat menyusun langkah kerja. Kemudian dilanjutkan dengan merancang *state diagram* berdasarkan langkah kerja dari tahapan sebelumnya. *State diagram* yang telah dirancang akan dikonversi ke dalam *ladder diagram* agar dapat diprogram pada PLC.

3.1 Perumusan Sistem *Dual Conveyor*

Plant dual conveyor yang digunakan dalam penelitian ini telah dirancang untuk melakukan fungsi khusus, yaitu melakukan proses seleksi dan perakitan benda kerja yang terdiri dari material plastik dan metal. Proses-proses tersebut dapat berlangsung karena sistem *dual conveyor* dilengkapi dengan sejumlah sensor dan aktuator tertentu. Sensor beroperasi sebagai *input* pada sistem dan aktuator berperan sebagai *output* sistem. Setelah itu dilanjutkan dengan membuat langkah kerja dari sistem dari awal hingga akhir.

3.1.1 I/O Sistem

Gambar 3.2 menunjukkan bagian I/O sistem *dual conveyor* yang digunakan pada proses kerja.



Gambar 3.2 Input/Output Sistem

Keterangan bagian *input* dan *output* (I/O) pada sistem *dual conveyor* beserta fungsinya dapat dilihat pada Tabel 3.1 (bagian *input*) dan Tabel 3.2 (bagian *output*).

Tabel 3.1 *Input* Sistem

No	<i>Input</i>	Keterangan	Fungsi
1	Start	Tombol Mulai	Untuk memulai proses keseluruhan.
2	Stop	Tombol Berhenti	Untuk menghentikan proses keseluruhan.
3	OPT1	Sensor Opto 1	Mendeteksi kehadiran benda kerja sebelum proses pengukuran ketinggian.
4	Hmin	Tinggi Minimum	Menjadi indikator ketebalan benda kerja minimum.
5	Hmax	Tinggi Maksimum	Menjadi indikator ketebalan benda kerja maksimum.
6	HTDC	<i>Height Top Dead Center</i>	Batas atas pergerakan motor (HMDRV) dalam mengukur ketinggian.
7	HBDC	<i>Height Bottom Dead Center</i>	Batas bawah pergerakan motor (HMDRV) dalam mengukur ketinggian.
8	OPT2	Sensor Opto 2	Mendeteksi kehadiran benda kerja ketika berada di induktif transduser.
9	IND	Induktif <i>Transducer</i>	Mendeteksi jenis material dari benda kerja.
10	OPT3	Sensor Opto 3	Mendeteksi kehadiran benda kerja pada chute 2 (untuk benda kerja logam).
11	OPT4	Sensor Opto 4	Mendeteksi kehadiran benda kerja pada chute 1 (untuk benda kerja plastik).
12	OPT5	Sensor Opto 5	Mendeteksi kehadiran benda kerja pada posisi akhir proses perakitan (untuk benda kerja logam).
13	OPT6	Sensor Opto 6	Mendeteksi kehadiran benda kerja pada posisi akhir proses perakitan (untuk benda kerja plastik).

Tabel 3.2 Output Sistem

No	Output	Keterangan	Fungsi
1	CONV1	<i>Conveyor 1</i> (atas)	Membawa benda kerja saat dimulainya proses.
2	DISP1	<i>Dispenser 1</i> (Washer)	Mengeluarkan benda kerja washer menuju <i>conveyor 1</i> .
3	HMDRV	<i>Height Motor Drive</i>	Motor pengukur ketinggian, untuk menggerakkan <i>plunger</i> (ujung pengukur) ke atas dan ke bawah saat mengukur ketebalan benda kerja.
4	SOL2	<i>Flipper 1</i>	Menyortir benda kerja material plastik ke <i>chute 1</i> .
5	SOL3	<i>Flipper 2</i>	Menyortir benda kerja material logam ke <i>chute 2</i> .
6	DISP2	<i>Dispenser 2</i> (Peg)	Mengeluarkan benda kerja peg menuju <i>conveyor 2</i> .
7	CONV2	<i>Conveyor 2</i> (bawah)	Membawa benda kerja peg untuk melakukan proses perakitan pada <i>washer</i> hingga selesai.
8	CHUT1	<i>Chute 1</i>	Mengeluarkan benda kerja <i>washer</i> yang standar (plastik) untuk proses perakitan.
9	CHUT2	<i>Chute 2</i>	Mengeluarkan benda kerja <i>washer</i> yang standar (logam) untuk proses perakitan.
10	SOL4	<i>Flipper 3</i>	Menyortir benda kerja logam yang sudah dirakit ke jalur dimana <i>Opto 5</i> berada.

3.1.2 Langkah Kerja Sistem

Setelah mengetahui semua bagian dan fungsi dari I/O sistem, langkah berikutnya adalah menyusun langkah atau proses kerja dari *dual conveyor* yang meliputi proses seleksi dan perakitan. Pada penyusunannya, penulis memecah proses kerja *dual conveyor* menjadi beberapa sub proses untuk memudahkan perancangan *state diagram* dan konstruksi *ladder* nantinya. Berikut ini dijelaskan kejadian dari setiap sub proses :

1. Sub Proses 1

Proses ini dimulai dari ditekannya tombol *start* sampai benda kerja terdeteksi oleh sensor *Opto 1* (Opt1). Bagian ini dibagi menjadi beberapa langkah :

- Saat tombol *start* ditekan, motor pengukur ketinggian (HMDRV) akan aktif. HMDRV akan menggerakkan *plunger* ke atas hingga mencapai batas atas ketinggian dan terdeteksi oleh sensor HTDC.
- Ketika HTDC aktif, HMDRV akan berhenti (nonaktif) dan dalam waktu yang bersamaan *dispenser* (DISP1) dan *conveyor* bagian atas (CONV1) juga aktif. *Dispenser* akan mengeluarkan *part washer* dan dibawa oleh *conveyor* menuju *Opto 1*.
- Kehadiran benda kerja yang terdeteksi oleh Opt1 akan membuat DISP1 dan CONV1 dinonaktifkan. *Solenoid* pada *dispenser* kembali ke posisi semula dan *conveyor* berhenti.

Tujuan dari menaikkan *plunger* ke posisi atas adalah untuk mencegah terjadinya tabrakan antara benda kerja dengan *plunger* tersebut sebelum proses pengukuran ketinggian benda kerja dilakukan. Hal itu mungkin akan terjadi apabila proses kerja sistem diulang dari awal setelah proses sebelumnya selesai (benda kerja mengalami *reject* atau setelah proses perakitan).

Bayangkan ketika proses kerja berulang dari awal lagi tapi posisi *plunger* tidak dinaikkan ke posisi atas (akibat dari pengukuran ketinggian pada proses sebelumnya), maka benda kerja akan bertabrakan dengan *plunger* yang posisinya masih di bawah sehingga proses pengukuran ketinggian menjadi terganggu.

2. Sub Proses 2

Sebagai lanjutan dari proses 1, proses 2 berawal dari aktifnya sensor *Opto 1* sampai benda kerja terdeteksi oleh sensor *Opto 2* (Opt2).

- Ketika benda kerja terdeteksi oleh Opt1 maka *timer 1* (Timer1) dan CONV1 daktifkan. *Conveyor* akan berhenti selama 1 detik.
- Setelah 1 detik, *timer 2* (Timer2) dan CONV1 akan aktif. *Conveyor* berjalan membawa benda kerja menuju tepat di bawah *plunger* (ujung pengukur ketinggian) dalam waktu 1,4 detik.

- Setelah 1,4 detik maka HMDRV akan aktif dan CONV1 kembali nonaktif. *Conveyor* berhenti dan benda kerja berada tepat di bawah *plunger*. HMDRV akan menggerakkan *plunger* untuk melakukan proses pengukuran ketinggian benda kerja.
- HMDRV bergerak hingga mencapai sensor HTDC. Aktifnya HTDC membuat HMDRV nonaktif dan CONV1 aktif. Setelah selesai mengukur ketinggian maka HMDRV dihentikan dan *conveyor* berjalan lagi membawa benda kerja menuju *Opto 2*.
- Ketika kehadiran benda kerja terdeteksi oleh Opt2, CONV1 akan dinonaktifkan. *Conveyor* berhenti agar sensor induktif dapat mendeteksi material dari benda kerja.

Pada proses 2 terdapat dua *input* tambahan pada sistem yang akan digunakan pada proses 3. *Input* yang pertama adalah *input* yang berperan sebagai indikator atau penanda bahwa ketinggian benda kerja sudah standar (disebut dengan “Flag1”). *Input* yang kedua adalah *input* yang berperan sebagai indikator bahwa benda kerja standar yang dideteksi berbahan logam (disebut dengan “Flag2”). Perlu diingat bahwa penamaan “Flag” yang dimaksud oleh penulis hanya sebagai tanda/indikator saja dan tidak berhubungan dengan kata “Flag” yang mengacu pada memori PLC.

3. Sub Proses 3

Proses ini terbagi menjadi dua yaitu proses untuk benda kerja dengan material logam dan proses untuk benda kerja dengan material plastik. Proses bermula dari aktifnya sensor *Opto 2* sampai benda kerja terdeteksi oleh ***Opto 3*** (untuk benda kerja **logam**) dan ***Opto 4*** (untuk benda kerja **plastik**).

- Untuk benda kerja **logam** maka **Opt2, Flag1 & Flag2** (sebagai *input*) akan mengaktifkan *timer 3* (**Timer3**). Untuk benda kerja **plastik** maka *input*-nya hanya **Opt2 & Flag1** saja dan akan mengaktifkan *timer 5* (**Timer5**). Timer3 maupun Timer5 berfungsi untuk menghentikan *conveyor* selama 1 detik.
- Setelah 1 detik, maka CONV1, **Sol3 & timer 4** (**Timer4**) akan aktif untuk benda kerja **logam**. Untuk benda kerja **plastik**, maka CONV1, **Sol2 & timer 6** (**Timer6**) akan aktif. *Conveyor* akan membawa benda kerja standar ke tempat penyimpanan sementara dengan bantuan Sol2 atau Sol3. Sol3 berfungsi untuk menyortir benda kerja logam ke tempat penyimpanan

sementara (dalam waktu 10 detik) sedangkan Sol2 berfungsi untuk menyortir benda kerja plastik ke tempat penyimpanan sementara (dalam waktu 7 detik).

- Setelah benda kerja masuk ke tempat penyimpanan sementara, CONV1 & Sol3 akan nonaktif dan **Chute2** akan aktif (untuk benda kerja **logam**). Untuk benda kerja **plastik** maka CONV1 & Sol2 akan nonaktif dan **Chute1** akan aktif. *Chute* berfungsi untuk menjatuhkan benda kerja dari tempat penyimpanan sementara ke tempat perakitan.
- Saat benda kerja logam terdeteksi oleh Opt3 maka Chute2 akan dinonaktifkan sedangkan benda kerja plastik yang terdeteksi oleh Opt4 akan menonaktifkan Chute1. Kehadiran benda kerja di tempat perakitan akan dideteksi oleh sensor *Opto* sebelum dilakukan proses perakitan dan memulai proses berikutnya.

4. Sub Proses 4

Melanjutkan dari proses sebelumnya, proses 4 dimulai ketika benda kerja terdeteksi oleh sensor *Opto* 3 atau *Opto* 4 sampai benda kerja terdeteksi oleh *Opto* 5 atau *Opto* 6.

- Ketika benda kerja **logam** terdeteksi oleh Opt3 maka **DISP2, CONV2 & Sol4** akan aktif. Untuk benda kerja **plastik** akan dideteksi oleh Opt4 sehingga **DISP2 & CONV2** saja yang aktif. *Dispenser* akan mengeluarkan benda kerja *peg* dan dibawa oleh *conveyor* untuk melakukan proses perakitan. Setelah proses perakitan selesai, benda kerja akan disortir oleh Sol4 menuju salah satu jalur sesuai jenis materialnya dan disimpan.
- Jika benda kerja **logam** yang sudah dirakit mencapai **Opt5** maka DISP2, CONV2 & Sol4 dinonaktifkan. Jika benda kerja **plastik** yang sudah dirakit mencapai **Opt6** maka DISP2 & CONV2 dinonaktifkan.

5. Sub Proses 5

Proses ini berfungsi untuk mengulangi proses kerja sistem *dual conveyor* dari awal jika benda kerja dinyatakan *reject* atau sudah selesai dilakukan proses perakitan.

- Jika *reject* maka **Opt2 & Flag1** menjadi *input* untuk mengaktifkan DISP1 & CONV1. Hal ini bertujuan untuk mengulangi proses segera setelah benda kerja yang diukur

tidak standar. Akan tetapi kondisi *input* dari Flag1 akan dibalik logika *ladder*-nya (dari NO menjadi NC) agar dapat membedakan proses yang *reject* atau bukan. Jika **Opt5** atau **Opt6** aktif maka DISP & CONV1 juga akan diaktifkan. Tujuannya juga untuk mengulangi proses dari awal tapi dengan kondisi benda kerja sudah dirakit.

- Saat proses berulang dari awal lagi sampai benda kerja dideteksi oleh Opt1 maka DISP1 & CONV1 kembali dinonaktifkan. Proses akan berulang terus sampai tombol *stop* ditekan.

3.2 Perancangan State Diagram

Pada penelitian ini perancangan *state diagram* disusun sesuai langkah-langkah metode *state diagram* pada dasar teori. Dimulai dengan mendefinisikan bit *input/output* yang akan digunakan pada sistem *dual conveyor*. Selain informasi tersebut, pada Tabel 3.3 juga terdapat alamat *input/output* untuk pemrograman PLC.

Tabel 3.3 Bit I/O Sistem

Bit	INPUT		OUTPUT	
	Keterangan	Simbol	Keterangan	Simbol
1	Relay Start	X ₁	HMDRV	Z ₁
2	HTDC	X ₂	DISP1	Z ₂
3	Opt1	X ₃	CONV1	Z ₃
4	T1 Contact	X ₄	T1 Coil	Z ₄
5	T2 Contact	X ₅	T2 Coil	Z ₅
6	Opt2	X ₆	T3 Coil	Z ₆
7	Flag1	X ₇	Sol3	Z ₇
8	Flag2	X ₈	T4 Coil	Z ₈
9	T3 Contact	X ₉	Chute2	Z ₉
10	T4 Contact	X ₁₀	T5 Coil	Z ₁₀
11	Opt3	X ₁₁	Sol2	Z ₁₁

Bit	INPUT		OUTPUT	
	Keterangan	Simbol	Keterangan	Simbol
12	T5 Contact	X_{12}	T6 Coil	Z_{12}
13	T6 Contact	X_{13}	Chute1	Z_{13}
14	Opt4	X_{14}	DISP2	Z_{14}
15	Opt5	X_{15}	CONV2	Z_{15}
16	Opt6	X_{16}	Sol4	Z_{16}

Kejadian-kejadian pada sistem *dual conveyor* juga diterjemahkan ke dalam tabel sebelum menyusun *state diagram* (I/O) agar memudahkan dalam penyusunannya. Tabel 3.4 menunjukkan kondisi *input/output* yang terjadi pada setiap *state*, mulai dari sub proses pertama hingga terakhir.

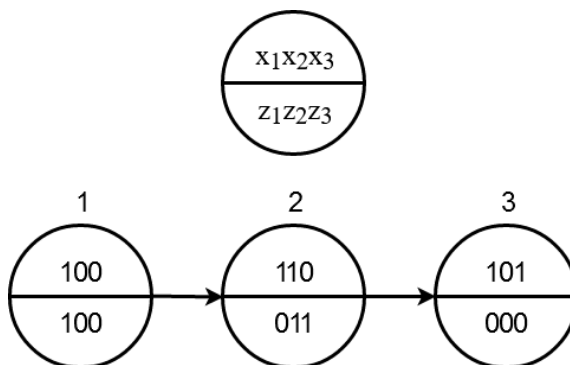
Tabel 3.4 Urutan *State Diagram* (I/O)

Sub Proses	State	Kondisi Input	Kondisi Output
1	1	Relay Start on	HMDRV on
	2	HTDC on	HMDRV off, DISP1 on, CONV1 on
	3	Opt1 on	DISP1 off, CONV1 off
2	1	Opt1 on	T1 Coil on
	2	T1 Contact on	T2 Coil on, CONV1 on
	3	T2 Contact on	HMDRV on, CONV1 off
	4	HTDC on	HMDRV off, CONV1 on
	5	Opt2 on	CONV1 off
3	Metal	Opt2, Flag1 & Flag2 on	T3 Coil on
		T3 Contact on	CONV1 on, Sol3 on, T4 Coil on
		T4 Contact on	CONV1 off, Sol3 off, Chute2 on
		Opt3 on	Chute2 off
	Plastik	Opt2 & Flag1 on	T5 Coil on
		T5 Contact on	CONV1 on, Sol2 on, T6 Coil on
		T6 Contact on	CONV1 off, Sol2 off, Chute1 on
		Opt4 on	Chute1 off

Sub Proses		State	Kondisi Input	Kondisi Output
4	Metal	1	Opt3 on	DISP2 on, CONV2 on, Sol4 on
		2	Opt5 on	DISP2 off, CONV2 off, Sol4 off
	Plastik	1	Opt4 on	DISP2 on, CONV2 on
		2	Opt6 on	DISP2 off, CONV2 off
5		1	Opt2 on or;	DISP1 on, CONV1 on
		2	Opt5 on or;	
		3	Opt6 on	
		4	Opt1 on	DISP1 off, CONV1 off

3.2.1 Penyusunan *State Diagram (I/O)*

Pada tahapan ini, informasi pada Tabel 3.4 disusun satu per satu ke dalam *state diagram (I/O)* berdasarkan masing-masing sub proses (lihat Gambar 3.3). Selain itu juga terdapat informasi susunan bit I/O setiap *state* yang dapat dilihat pada Tabel 3.5. *State diagram (I/O)* dari setiap sub proses dapat dilihat pada Lampiran.



Gambar 3.3 *State Diagram (I/O)* Sub Proses 1

3.2.2 Penyusunan *Primitive Flow Table*

Informasi dari *state diagram* (I/O) secara keseluruhan pada setiap sub proses digunakan untuk menyusun *primitive flow table* sistem *dual conveyor*. Tabel 3.5 menampilkan *primitive flow table* dari sub proses 1. *Primitive flow table* dari setiap sub proses dapat dilihat pada Lampiran.

Tabel 3.5 *Primitive Flow Table* Sub Proses 1

Row	Inputs $x_1x_2x_3$			Outputs $z_1z_2z_3$		
	100	110	101	z_1	z_2	z_3
1	1	2	-	1	0	0
2	-	2	3	0	1	1
3	-	-	3	0	0	0

3.2.3 Penyusunan *Merged Flow Table*

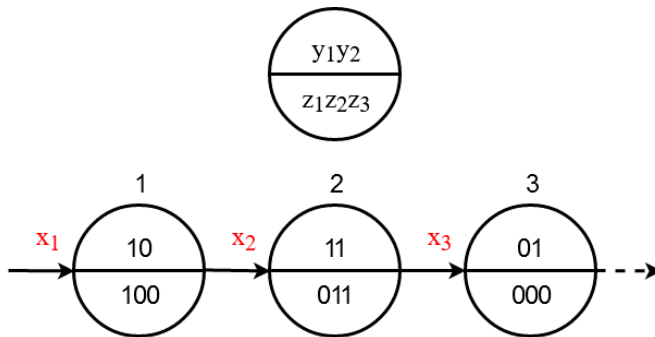
Dalam penyusunan *merged flow table* sub proses 1, tidak ada baris yang dapat digabungkan karena kombinasi bit *output*-nya tidak ada yang sama. Hasil *merged flow table* sub proses 1 ditunjukkan oleh Tabel 3.6. *Merged flow table* dari setiap sub proses dapat dilihat pada Lampiran.

Tabel 3.6 *Merged Flow Table* Sub Proses 1

Row	Inputs $x_1x_2x_3$			Outputs $z_1z_2z_3$			System's	
	100	110	101	z_1	z_2	z_3	y_1	y_2
1	1	2	-	1	0	0	1	0
2	-	2	3	0	1	1	1	1
3	-	-	3	0	0	0	0	1

3.2.4 Penyusunan *State Diagram* (R/O)

Pada langkah sebelumnya telah didapatkan *merged flow table*. Informasi di dalam tabel akan digunakan untuk menyusun *state diagram* (R/O) yang memiliki susunan *relay/output* pada setiap *state*. Gambar 3.4 menunjukkan hasil penyusunan *state diagram* (R/O) untuk sub proses 1. *State diagram* (R/O) dari setiap sub proses dapat dilihat pada Lampiran.

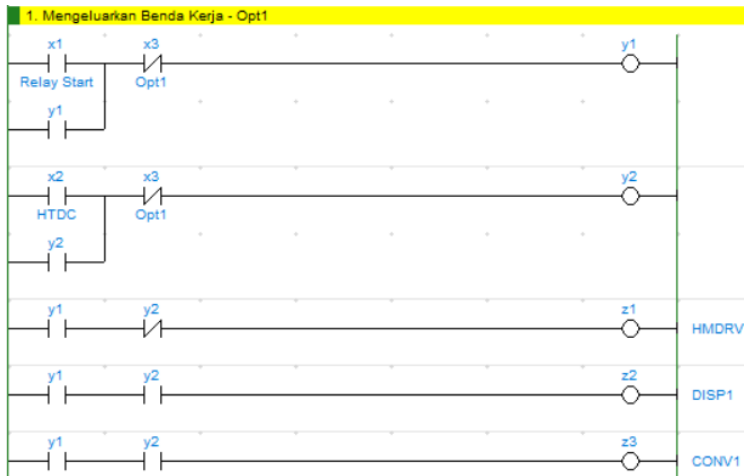


Gambar 3.4 *State Diagram (R/O) Sub Proses 1*

3.3 Konstruksi *State (R/O)-Ladder*

Dari hasil perancangan *state diagram* didapatkan penggunaan *relay* sebanyak 12 buah dan *output* sebanyak 16 buah. Oleh karena itu dilakukan konstruksi *ladder* untuk 12 *rung relay* (y_1 hingga y_{12}) dan 16 *rung output* (z_1 hingga z_{16}). Akan tetapi terdapat tambahan *rung* sebanyak 4 buah untuk pendukung proses kerja masing-masing untuk tombol *Start*, *Stop*, *Flag1* dan *Flag2*. Hasil konstruksi *ladder diagram* untuk proses kerja *dual conveyor* adalah sebanyak 32 *rung* jika ditotalkan.

Untuk tombol “**Start**”, tombol “**Stop**”, dan “**IND**” pada *ladder diagram*, digunakan *contact NC* agar program dapat berjalan karena konfigurasi *pin* (kaki) positif dan negatif dari *input* fisik terbalik. Gambar 3.5 menunjukkan hasil konstruksi *ladder diagram* untuk sub proses 1. Hasil konstruksi untuk setiap sub proses dapat dilihat pada Lampiran.



Gambar 3.5 Konstruksi *Ladder* untuk Sub Proses 1

[Halaman ini sengaja dikosongkan]

BAB 4

PENGUJIAN DAN ANALISA

Ladder diagram yang telah dikonstruksi akan disimulasikan terlebih dahulu ke PLC sebelum diuji pada *plant dual conveyor*. Agar sistem dapat bekerja saat melakukan pengujian maka *power supply*, PLC, dan *plant* harus terhubung satu sama lain sehingga dibutuhkan *wiring* antar perangkat-perangkat tersebut.

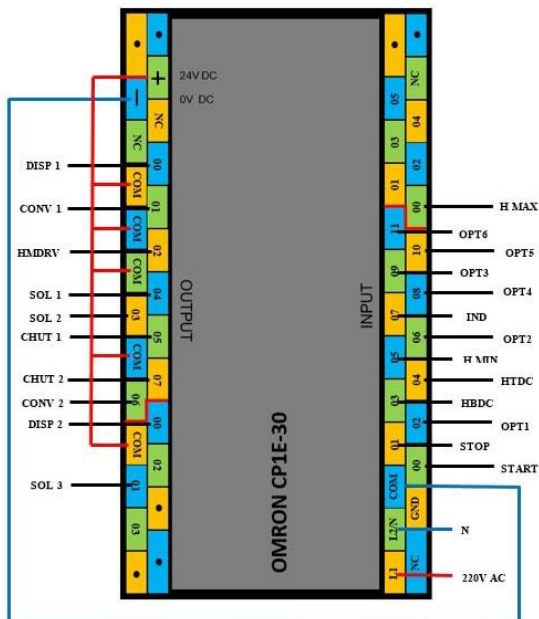
4.1 Pengkabelan (*Wiring*)

Untuk dapat melakukan komunikasi antara kontroler (PLC) dan *plant (dual conveyor)* diperlukan koneksi pada keduanya. Cara untuk menghubungkan keduanya adalah dengan melakukan *wiring* antara PLC dengan *Interface PCB* yang ada pada *plant*. *Interface PCB* memiliki terminal untuk setiap I/O sehingga memudahkan pengguna melakukan *wiring* ke PLC. Pemasangan kabel di PLC harus menyesuaikan alamat pada Tabel 4.1. *Wiring* yang dimaksud hanya menghubungkan *input/output* fisik ke PLC seperti pada Gambar 4.1. *Input/output* seperti “Relay Start” tidak perlu di-*wiring* karena menggunakan alamat memori dari *relay* internal (bersifat virtual).

Tabel 4.1 Alamat I/O PLC

No.	INPUT		OUTPUT	
	Keterangan	Alamat	Keterangan	Alamat
1	Relay Start	10.00	HMDRV	100.02
2	HTDC	0.04	DISP1	100.00
3	Opt1	0.02	CONV1	100.01
4	T1 Contact	T001	T1 Coil	T001
5	T2 Contact	T002	T2 Coil	T002
6	Opt2	0.06	T3 Coil	T003
7	Flag1	11.06	Sol3	100.04
8	Flag2	11.07	T4 Coil	T004
9	T3 Contact	T003	Chute2	100.06

No.	INPUT		OUTPUT	
	Keterangan	Alamat	Keterangan	Alamat
10	T4 Contact	T004	T5 Coil	T005
11	Opt3	0.09	Sol2	100.03
12	T5 Contact	T005	T6 Coil	T006
13	T6 Contact	T006	Chute1	100.05
14	Opt4	0.08	DISP2	101.00
15	Opt5	0.10	CONV2	100.07
16	Opt6	0.11	Sol4	101.01



Gambar 4.1 Pengkabelan pada PLC

Pastikan seluruh kabel terpasang dengan benar, bisa menggunakan AVO meter untuk tes koneksi dari PLC menuju antarmuka PCB, kemudian tes koneksi dari antarmuka menuju sensor atau aktuatur jika diperlukan. Proses tes ini bisa dilakukan juga untuk menyelesaikan masalah jika terjadi ketidaksesuaian antara program yang sedang berjalan pada *ladder* diagram di PLC dengan kondisi aktual pada *plant dua conveyor*. Program pada komputer ditransfer terlebih dahulu ke PLC.

4.2 Pengujian Sistem

Pengujian pada sistem dilakukan untuk mengetahui apakah implementasi berjalan sesuai dengan yang diharapkan. Penulis melakukan pengujian dengan menggunakan benda kerja yang standar dengan menjalankan sistem dari proses seleksi hingga proses perakitan selesai. Pengambilan data dilakukan menggunakan benda kerja yang standar untuk mengetahui berapa lama waktu dari satu kali (siklus) proses seleksi dan perakitan terjadi. Data pengujian ditunjukkan oleh Tabel 4.2.

a. Pengeluaran Benda kerja

Akan terjadi setelah tombol start (warna hijau) ditekan dan proses kalibrasi sensor ketinggian dilakukan.

b. Pengukuran Ketebalan

Dilakukan setelah benda kerja 1 terdeteksi Opt1 dan berjalan selama 1,4 detik sehingga berhenti tepat dibawah plunger.

c. Pembuangan Benda Kerja 1

Benda kerja 1 akan dibuang jika ketebalan tidak memenuhi standar yang diinginkan.

d. Seleksi material

Jika tebal benda kerja sesuai, maka akan diproses lanjut untuk diseleksi jenis material dari benda kerja 1 oleh sensor induktif.

e. Pemisahan Benda Kerja 1

Benda kerja akan dipisahkan berdasarkan jenis material yang terdeteksi. Bahan plastik akan diarahkan menuju peluncur (*chute1*) dan bahan logam akan diarahkan menuju peluncur (*chute2*).

f. Perakitan Benda Kerja

Jika benda kerja telah berada pada Peluncur dan telah dipindahkan oleh *dispenser chute* 1 atau 2 akan dikeluarkan Benda kerja 2 dan digerakan oleh conveyor 2 untuk bergerak menuju lokasi perakitan.

g. Pemisahan Hasil Perakitan

Hasil perakitan kombinasi Plastik akan diarahkan menuju Opt6. Hasil perakitan kombinasi metal akan diarahkan menuju Opt5. Kedua hasil kombinasi ini dipisah oleh *flipper 3* yang berfungsi sebagai pengarah.

Tabel 4.2 Hasil Pengujian 10 Siklus

Data Pengujian Benda Kerja Standar (8 mm)		
Siklus ke-	Metal (detik)	Plastik (detik)
1	37.23	33.32
2	36.74	33.20
3	36.97	32.97
4	37.11	33.20
5	37.74	33.13
6	35.88	32.88
7	36.86	32.06
8	38.02	32.88
9	38.18	33.10
10	36.95	33.40
Waktu Rata-Rata 1 Siklus (detik)	37.168	33.014

Tabel 4.2 menunjukkan waktu pengujian untuk proses seleksi dan perakitan *part* metal maupun plastik dari awal hingga akhir. Waktu untuk proses kerja *part* plastik lebih cepat (33.014 detik) daripada *part* metal (37.168 detik). Hal tersebut dikarenakan benda kerja (untuk material plastik) yang bergerak ke “Chute1” lebih dekat jaraknya daripada ke “Chute2”.

BAB 5

KESIMPULAN DAN SARAN

Hasil penelitian yang telah dilakukan dan kendala yang dihadapi selama proses pengerjaan dimuat dalam kesimpulan dan saran.

5.1 Kesimpulan

Berdasarkan proses konstruksi *ladder diagram* yang telah dilakukan pada penelitian tugas akhir ini, dapat disimpulkan beberapa hal, antara lain :

1. Konstruksi *ladder diagram* dengan metode *State Diagram* untuk proses seleksi dan perakitan pada *Dual Conveyor* menghasilkan total 32 *rung ladder* dengan rincian 4 *rung* pendukung proses, 12 *rung relay* (untuk 12 buah *relay* yang dihasilkan) dan 16 *rung output*.
2. Terdapat 6 buah *Timer* dan 16 buah alamat memori.
3. Kapasitas memori untuk *file* program yang dihasilkan sebesar 3 KB.
4. Satu siklus proses seleksi dan perakitan benda kerja standar memiliki waktu rata-rata 37.168 detik untuk *part* metal dan 33.014 detik untuk *part* plastik.
5. Untuk sistem yang sangat kompleks, metode *State Diagram* tidak cocok di terapkan karena akan menghasilkan diagram *ladder* yang tidak optimal. Akan tetapi permasalahan tersebut dapat diatasi dengan membagi proses keseluruhan sistem menjadi sub-sub proses.

5.2 Saran

Beberapa saran yang perlu diberikan untuk penelitian selanjutnya antara lain :

1. Pada saat implementasi lebih baik menggunakan *power supply* yang sesuai dengan spesifikasi kebutuhan *plant*. Spesifikasi *power supply* yang tidak sesuai dapat mengganggu berjalannya proses pada *plant*.
2. Akan lebih baik jika proses dapat berjalan paralel dalam 1 siklus proses untuk mempercepat proses seleksi dan perakitan benda kerja. Misalnya *conveyor* atas dapat berjalan secara bersamaan dengan *conveyor* bawah tanpa harus menunggu selesainya proses perakitan.
3. Membuat tampilan atau *user interface* seperti HMI untuk mempermudah pengguna memantau proses yang terjadi pada *plant*.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

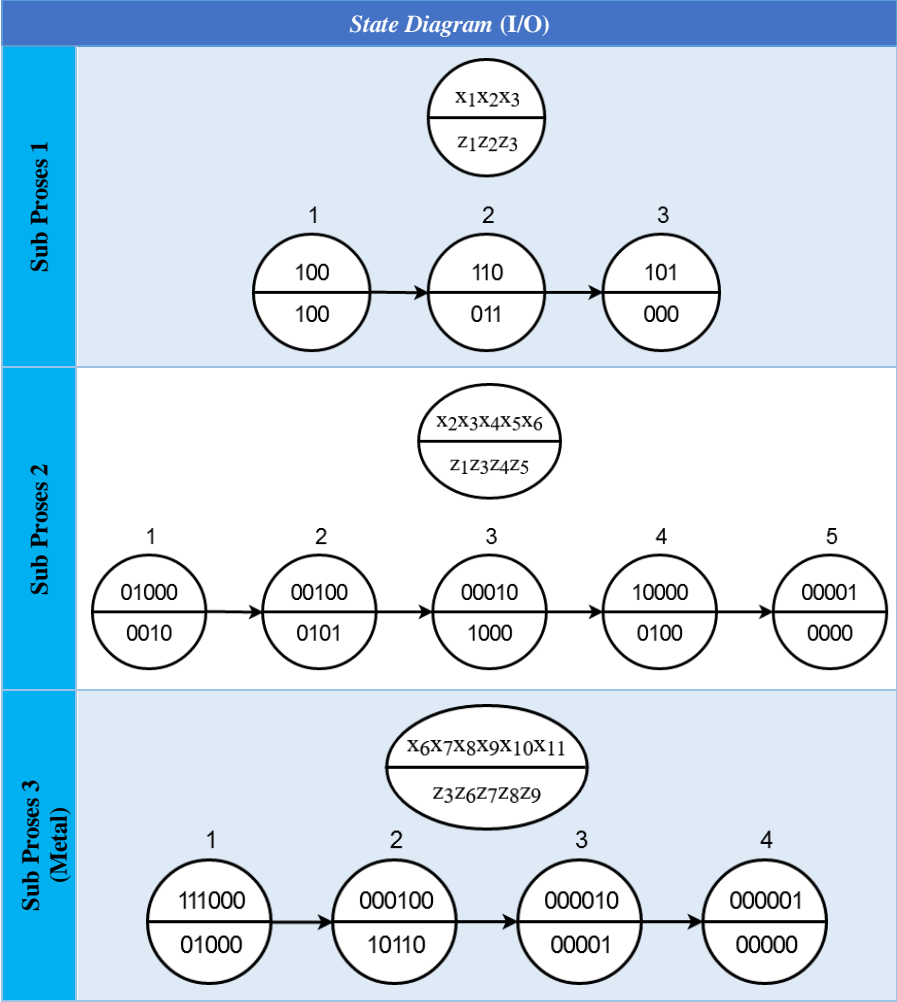
- [1] Anonim., *"Single Conveyor & Workcell Systems 34-001 to 34-004"*. Crowborough England: FI Ltd.
- [2] Anonim., "Linier Solenoid Actuator" <URL: http://www.electronics-tutorials.ws/io/io_6.html>, Mei, 2017.
- [3] Anonim., "What is a Photoelectric Sensor" <URL: <http://www.keyence.com/ss/products/sensor/sensorbasics/photoelectric/info/>>, Mei, 2017.
- [4] Menke, Henry., "Basic Operating Principle of an Inductive Proximity Sensor" <URL: <https://sensortech.wordpress.com/2014/03/05/basic-operating-principle-of-an-inductive-proximity-sensor/>>, Mei, 2017.
- [5] H. Wicaksono, PROGRAMMABLE LOGIC CONTROLLER Teori, Pemrograman dan Aplikasinya dalam Otomasi Sistem, Yogyakarta: GRAHA ILMU, 2009.
- [6] Anonim., *SYSMAC CPl/CPIE Introduction Manual.*, Omron.
- [7] D, Persen., *"Industrial Automation : Circuit Design and Components"*, Israel: John Wiley & Sons. 1989.

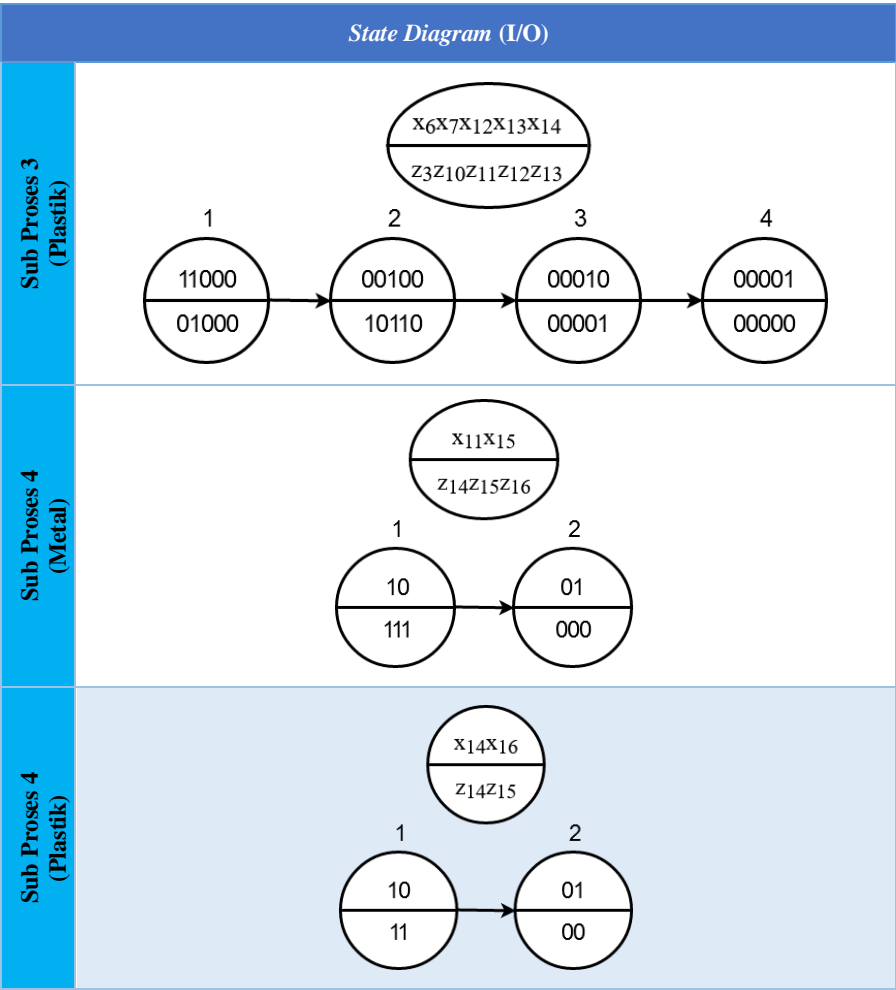
[Halaman ini sengaja dikosongkan]

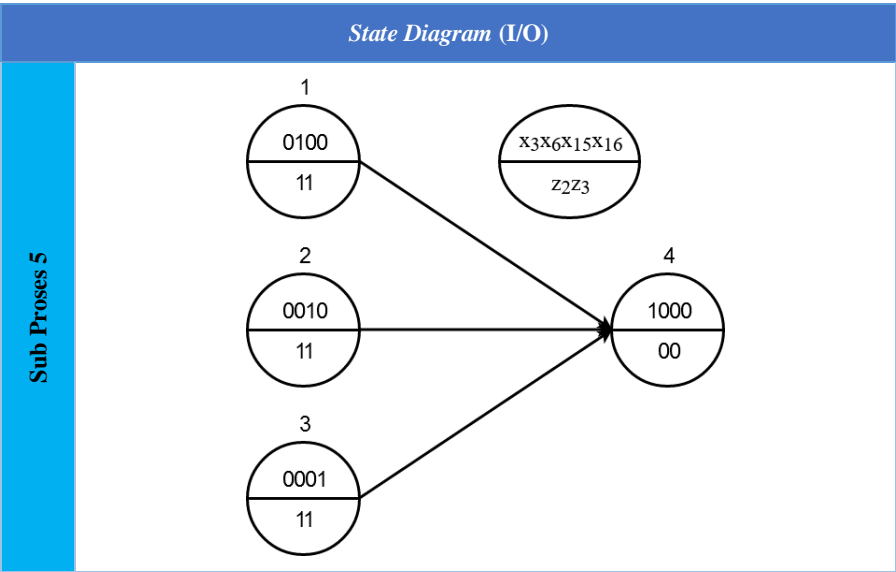
LAMPIRAN

A. State Diagram (I/O)

Tabel A1 State Diagram (I/O)







B. Primitive Flow Table

Tabel B1 Primitive Flow Table Sub Proses 1

Row	Inputs $x_1x_2x_3$			Outputs $z_1z_2z_3$		
	100	110	101	z_1	z_2	z_3
1	1	2	-	1	0	0
2	-	2	3	0	1	1
3	-	-	3	0	0	0

Tabel B2 Primitive Flow Table Sub Proses 2

Row	Inputs $x_2x_3x_4x_5x_6$					Outputs $z_1z_3z_4z_5$			
	01000	00100	00010	10000	00001	z_1	z_3	z_4	z_5
1	1	2	-	-	-	0	0	1	0
2	-	2	3	-	-	0	1	0	1
3	-	-	3	4	-	1	0	0	0
4	-	-	-	4	5	0	1	0	0
5	-	-	-	-	5	0	0	0	0

Tabel B3 *Primitive Flow Table* Sub Proses 3 (Metal)

Row	Inputs $x_6x_7x_8x_9x_{10}x_{11}$				Outputs $z_3z_6z_7z_8z_9$				
	11000	00100	000010	000001	z_3	z_6	z_7	z_8	z_9
1	1	2	-	-	0	1	0	0	0
2	-	2	3	-	1	0	1	1	0
3	-	-	3	4	0	0	0	0	1
4	-	-	-	4	0	0	0	0	0

Tabel B4 *Primitive Flow Table* Sub Proses 3 (Plastik)

Row	Inputs $x_6x_7x_{12}x_{13}x_{14}$				Outputs $z_3z_{10}z_{11}z_{12}z_{13}$				
	11000	00100	00010	00001	z_3	z_{10}	z_{11}	z_{12}	z_{13}
1	1	2	-	-	0	1	0	0	0
2	-	2	3	-	1	0	1	1	0
3	-	-	3	4	0	0	0	0	1
4	-	-	-	4	0	0	0	0	0

Tabel B5 *Primitive Flow Table* Sub Proses 4 (Metal)

Row	Inputs $x_{11}x_{15}$		Outputs $z_{14}z_{15}z_{16}$		
	10	01	z_{14}	z_{15}	z_{16}
1	1	2	1	1	1
2	-	2	0	0	0

Tabel B6 *Primitive Flow Table* Sub Proses 4 (Plastik)

Row	Inputs $x_{14}x_{16}$		Outputs $z_{14}z_{15}$	
	10	01	z_{14}	z_{15}
1	1	2	1	1
2	-	2	0	0

Tabel B7 *Primitive Flow Table* Sub Proses 5

Row	Inputs $x_3x_6x_{15}x_{16}$				Outputs z_2z_3	
	0100	0010	0001	1000	z_2	z_3
1	1	2	-	-	1	1
2	-	2	3	-	1	1
3	-	-	3	4	1	1
4	-	-	-	4	0	0

C. Merged Flow Table

Tabel C1 Merged Flow Table Sub Proses 1

Row	Inputs $x_1x_2x_3$			Outputs $z_1z_2z_3$			System's	
	100	110	101	z_1	z_2	z_3	y_1	y_2
1	1	2	-	1	0	0	1	0
2	-	2	3	0	1	1	1	1
3	-	-	3	0	0	0	0	1

Tabel C2 Merged Flow Table Sub Proses 2

Row	Inputs $x_2x_3x_4x_5x_6$					Outputs $z_1z_3z_4z_5$				System's		
	010 00	001 00	000 10	100 00	000 01	z_1	z_3	z_4	z_5	y_3	y_4	y_5
1	1	2	-	-	-	0	0	1	0	1	0	0
2	-	2	3	-	-	0	1	0	1	1	0	1
3	-	-	3	4	-	1	0	0	0	1	1	1
4	-	-	-	4	5	0	1	0	0	0	1	1
5	-	-	-	-	5	0	0	0	0	0	1	0

Tabel C3 Merged Flow Table Sub Proses 3 (Metal)

Row	Inputs $x_6x_7x_8x_9x_{10}x_{11}$				Outputs $z_3z_6z_7z_8z_9$					System's	
	111 000	000 100	000 010	000 001	z_3	z_6	z_7	z_8	z_9	y_6	y_7
1	1	2	-	-	0	1	0	0	0	1	0
2	-	2	3	-	1	0	1	1	0	1	1
3	-	-	3	4	0	0	0	0	1	0	1
4	-	-	-	4	0	0	0	0	0	0	0

Tabel C4 Merged Flow Table Sub Proses 3 (Plastik)

Row	Inputs $x_6x_7x_{12}x_{13}x_{14}$				Outputs $z_3z_{10}z_{11}z_{12}z_{13}$					System's	
	110 00	001 00	000 10	000 01	z_3	z_{10}	z_{11}	z_{12}	z_{13}	y_8	y_9
1	1	2	-	-	0	1	0	0	0	1	0
2	-	2	3	-	1	0	1	1	0	1	1
3	-	-	3	4	0	0	0	0	1	0	1
4	-	-	-	4	0	0	0	0	0	0	0

Tabel C5 Merged Flow Table Sub Proses 4 (Metal)

Row	Inputs $x_{11}x_{15}$		Outputs $z_{14}z_{15}z_{16}$			System's
	10	01	z_{14}	z_{15}	z_{16}	y_{10}
1	1	2	1	1	1	1
2	-	2	0	0	0	0

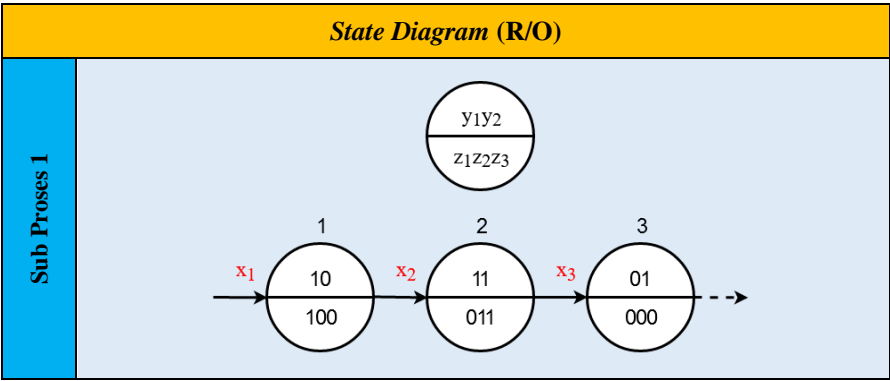
Tabel C6 Merged Flow Table Sub Proses 4 (Plastik)

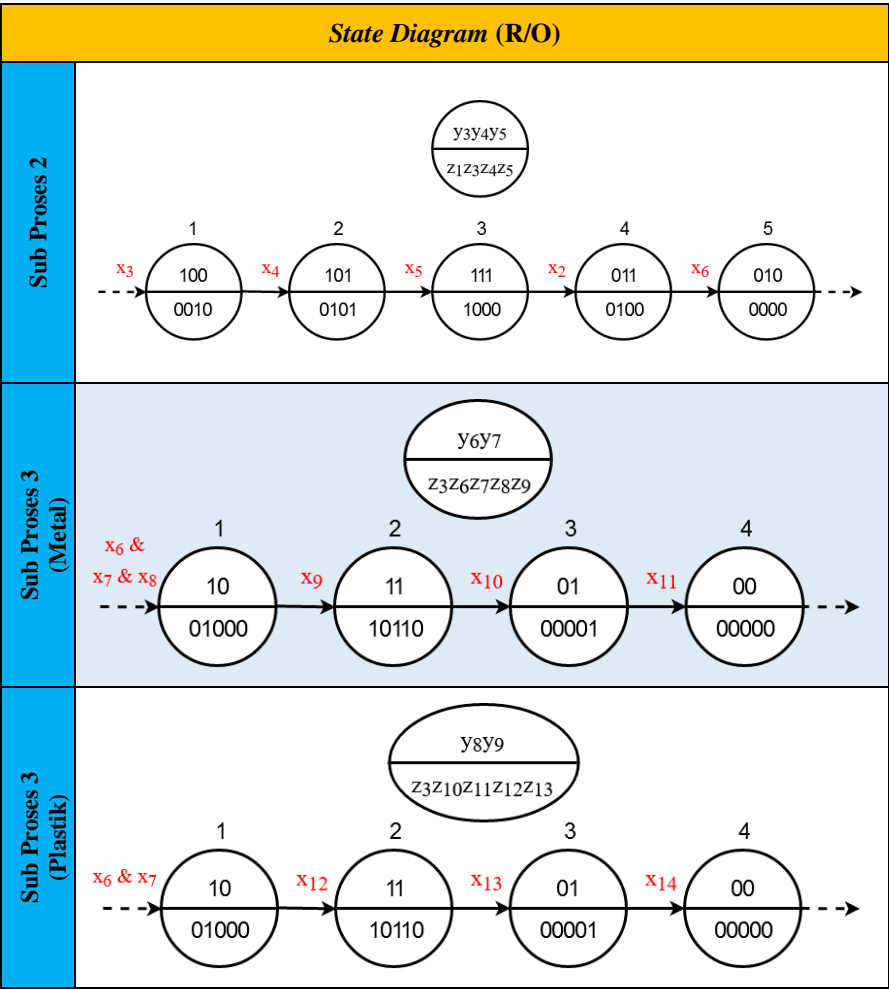
Row	Inputs $x_{14}x_{16}$		Outputs $z_{14}z_{15}$		System's
	10	01	z_{14}	z_{15}	y_{11}
1	1	2	1	1	1
2	-	2	0	0	0

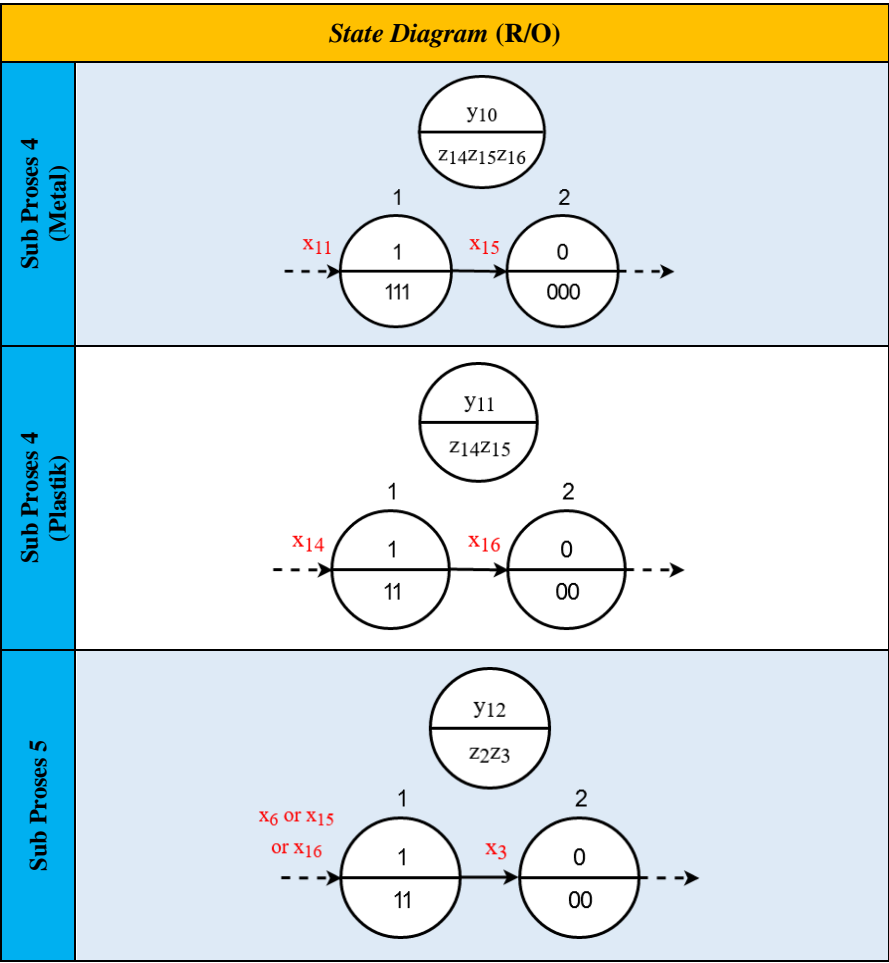
Tabel C7 Merged Flow Table Sub Proses 5

Row	Inputs $x_3x_6x_{15}x_{16}$				Outputs z_2z_3		System's
	0100	0010	0001	1000	z_2	z_3	y_{12}
1, 2, 3	1	2	3	4	1	1	1
4	-	-	-	4	0	0	0

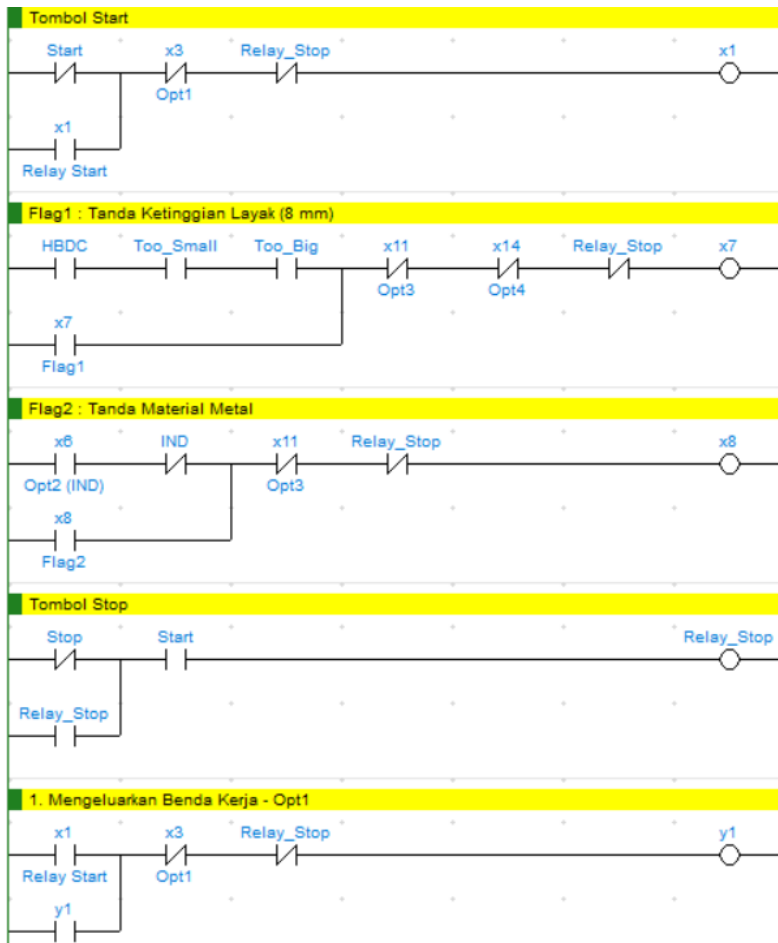
D. State Diagram (R/O)

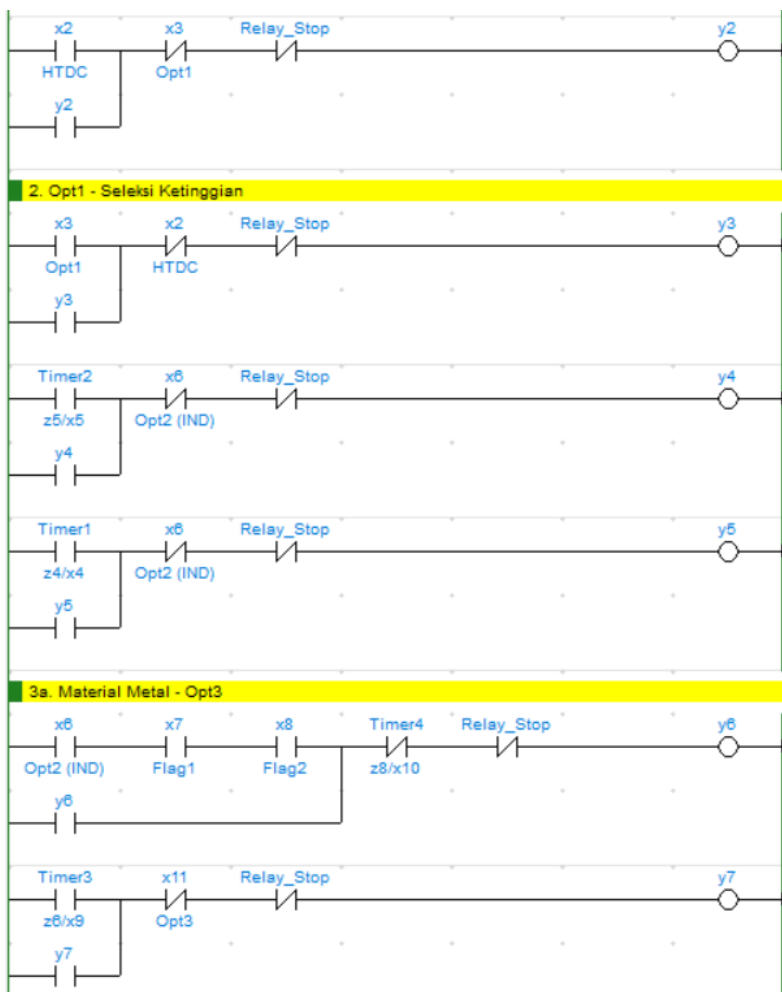




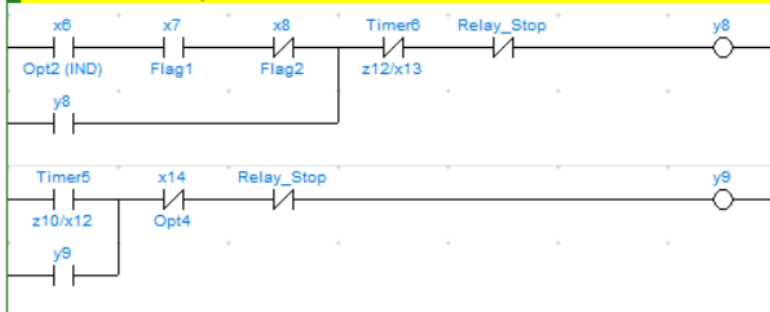


E. Ladder Diagram

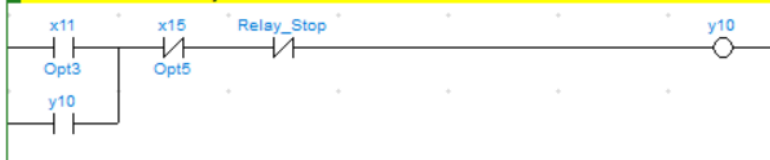




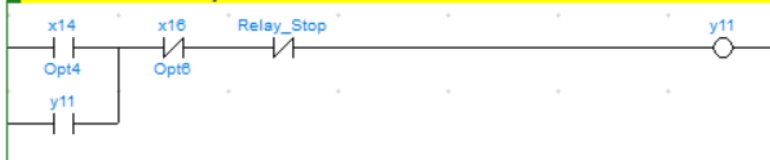
3b. Material Plastik - Opt4



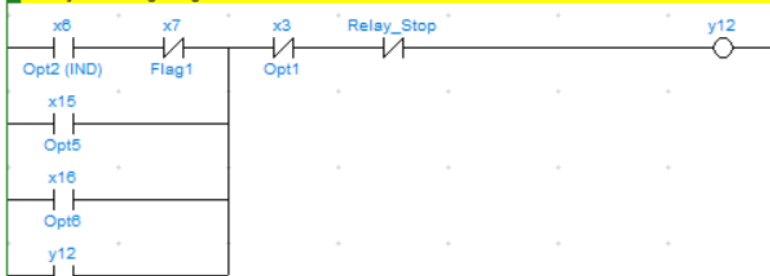
4a. Perakitan Benda Kerja Metal

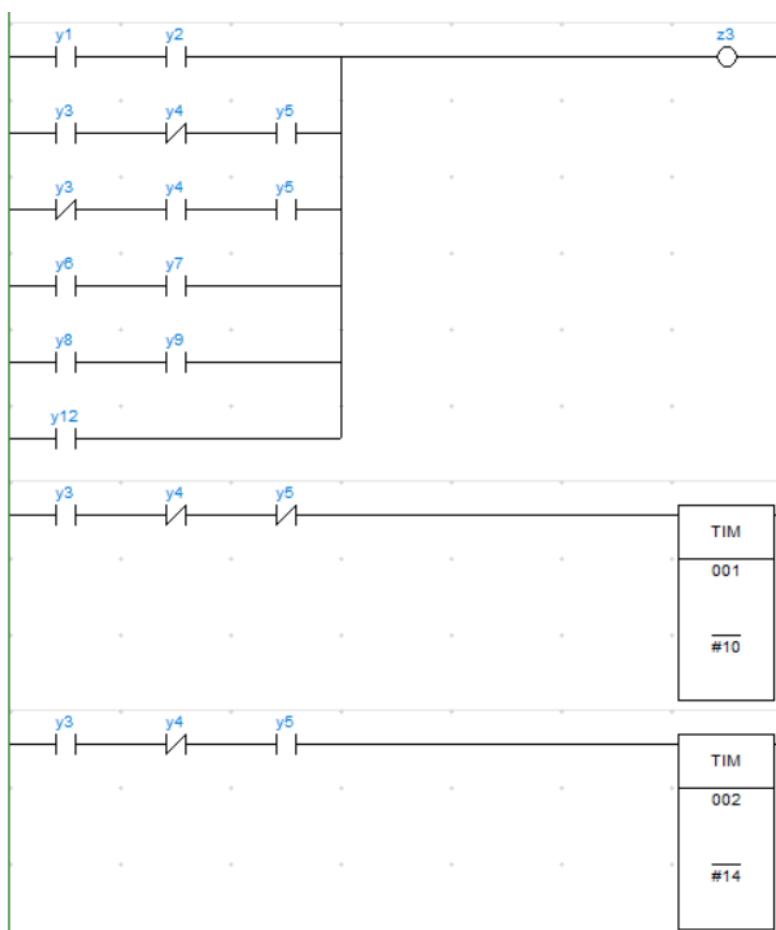


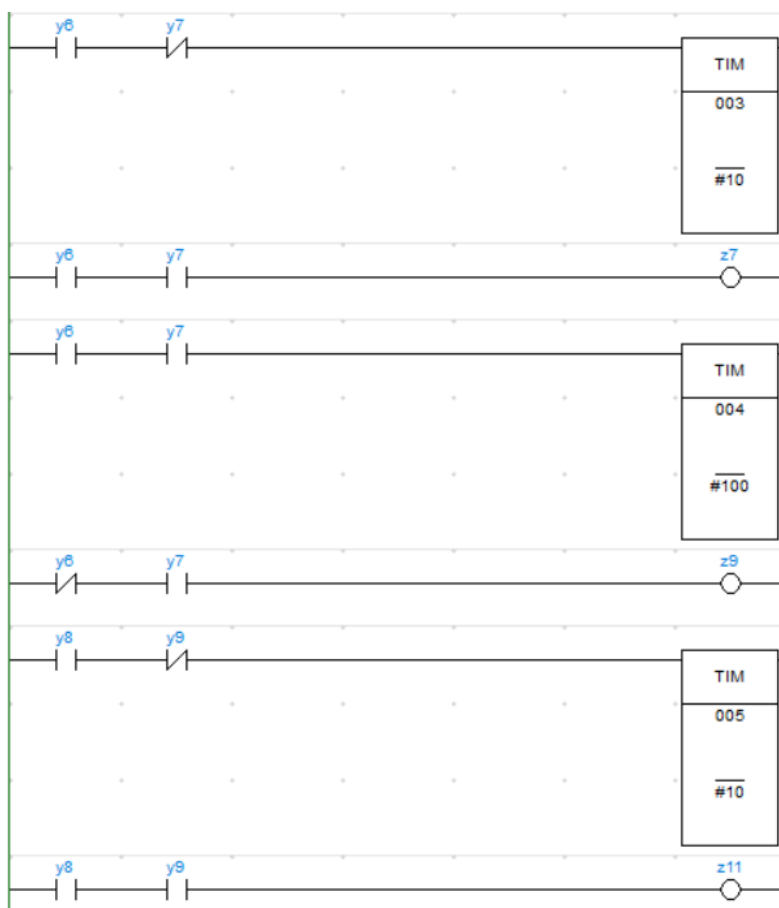
4b. Perakitan Benda Kerja Plastik

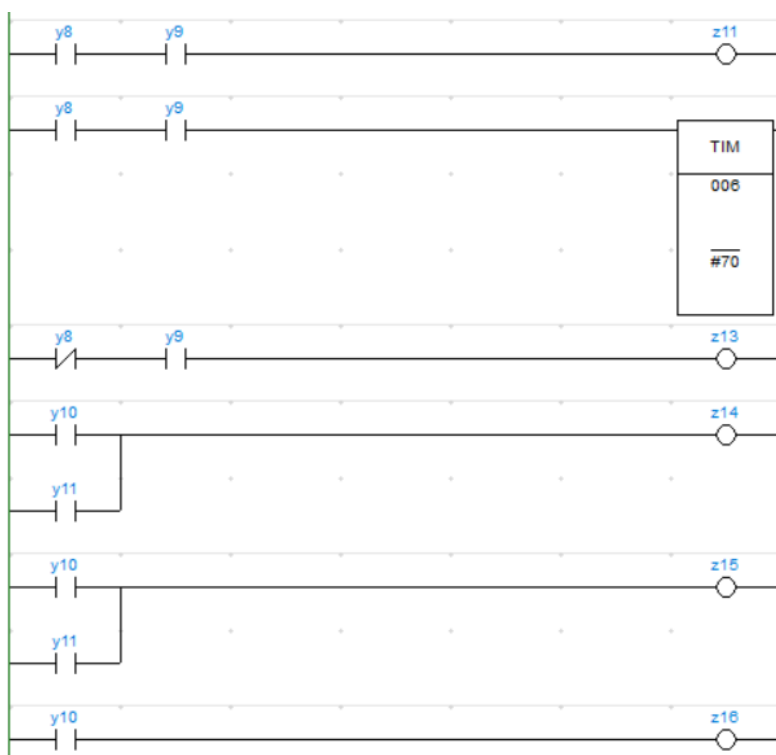


5. Reject & Pengulangan Proses









RIWAYAT HIDUP



Nicco, lahir di Singkawang pada tanggal 19 November 1992. Putra ketiga dari pasangan Bapak Lim Hui Boon dan Ibu Tjhung Bun Tjin. Setelah menempuh pendidikan formal di SD Ananda Batam, SMP Ananda Batam dan SMA Ananda Batam, penulis melanjutkan studi Diploma 3 jurusan Teknik Elektronika di Politeknik Negeri Batam dan lulus tahun 2014. Kemudian melanjutkan pendidikan S1 di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan mengambil Jurusan Teknik Elektro, Bidang Studi Teknik Sistem Pengaturan. Pada tanggal 13 Juli 2017, penulis mengikuti ujian Tugas Akhir untuk memperoleh gelar Sarjana Teknik.